arXiv:2305.03511v2 [cs.CL] 9 Sep 2024

用于非自回归神经机器翻译的双语共享潜在空间

DongNyeong Heo[1] 和 Heeyoul Choi[1*]

[1]计算机科学与电气工程系，韩东全球大学，庆尚北道浦项Handong-ro 558号，邮编37554，大韩民国。

*通讯作者。电子邮箱：hchoi@handong.edu；贡献作者：dnheo@handong.ac.kr；

摘要

与自回归神经机器翻译（AT）相比，非自回归神经机器翻译（非自回归翻译）能显著提升翻译速度，但代价是翻译质量下降。潜变量建模已成为弥合这一质量差距的一种有前景的方法，尤其有助于解决非自回归翻译中长期存在的多模态问题。在以往采用潜变量建模的研究中，研究者添加了一个辅助模型，用于估计以源句和目标句子为条件的潜变量后验分布。然而，这会带来若干缺点，例如潜变量中的信息提取冗余、参数量增加，以及模型倾向于忽略输入中的部分信息。本文提出一种新颖的潜变量建模方法，将双重重构视角与先进的分层潜变量建模相结合，并在不同语言之间采用共享中间潜在空间。理论上，这种潜变量建模可以缓解或避免上述缺点。在实验结果中，我们全面展示了所提方法能够推断出更优的潜变量，从而提升翻译质量。最后，在 WMT 等基准翻译任务中，我们证明，与以往的非自回归机器翻译基线（包括最先进水平的非自回归模型）相比，所提方法显著提升了翻译质量。

关键词：神经机器翻译、非自回归解码、潜变量建模、变分自编码器

1 引言

过去十年间，神经机器翻译（NMT）受到广泛关注 [1, 2], ，并通过自回归解码架构取得了显著进展。然而，这类架构受其序列化特性的限制：解码器必须根据先前生成的上下文，逐步生成每个目标词，因而限制了翻译速度。为解决这一限制，非自回归神经机器翻译（NAT）范式应运而生，旨在无需依赖先验上下文即可生成全部目标词。值得注意的是，与自回归模型（AT）相比，NAT模型显著提升了翻译速度 [3–11]。

然而，非自回归翻译（NAT）模型往往需要在翻译质量和翻译速度的提升之间做出权衡。导致翻译质量下降的主要因素是“多模态问题”：由于目标句子中所有词的分布共同作用，在确定目标句子的某个模式时会面临困难，尤其是在目标句子呈现多模态分布的情况下。产生这一问题的原因是，NAT模型在计算某个词的输出分布时，不以之前的目标词为条件。为解决这一问题，已有多种方法被提出，包括数据集预处理 [3, 12]，迭代解码过程 [4]，以及对目标函数的修改 [5]。此外，引入能够隐式表示目标句子某种模式的潜变量，是缓解这一问题的一种有前景的方法，通常能够提升性能 [6–8]。

潜变量建模的传统方法是在编码器-解码器架构之外加入辅助模型。该辅助模型用于近似潜变量在给定源句子 X 和目标句子 Y 条件下的后验分布 $q(Z|X, Y)$，其中潜变量为 Z。典型的编码器和解码器组件分别估计先验分布 $p(Z|X)$ 和似然 $p(Y|Z, X)$。随后，整个模型架构使用证据下界目标（ELBO）函数进行训练，这与变分自编码器（VAE）模型所采用的框架类似 [13]。

然而，这种潜变量建模方法有三个主要缺点。首先，在训练阶段，潜变量仅用于估计目标句子的似然分布。因此，潜变量可能会编码目标语言特定信息，而这些信息在测试时并不存在，对翻译任务也毫无帮助。其次，加入辅助后验模型会显著增加参数量，从而可能带来实际问题，例如难以将其部署为端侧应用。最后，辅助后验模型通常能够调节从每个输入中提取的信息比例。因此，后验模型可能会根据 ELBO 函数中的对数似然项或库尔贝克-莱布勒散度项，分别忽略源句子或目标句子。我们将这一最后的缺点称为单侧后验坍塌（OSPC）问题。存在 OSPC 问题时，模型在

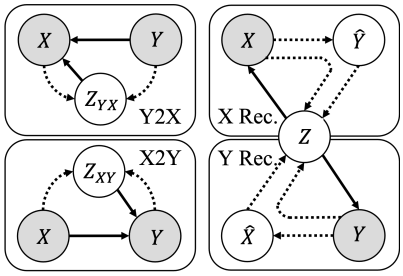


图 1 传统潜变量模型的图模型（左）。白色圆圈分别表示观测变量和潜变量，实线和虚线分别表示生成过程和推断过程。与先前方法不同，我们将整个翻译系统概念化为双重重构，并据此提出双层次潜变量模型，使中间潜变量 Z 由观测变量和更深层的潜变量（翻译句子）共同推断。此外，我们在两种语言的重构过程中共享中间潜变量。

测试阶段的表现可能会受到影响，这与第一个缺点类似，前提是模型在训练期间忽略了源句子。另一方面，如果模型在训练期间忽略了目标句子，其潜变量架构可能就失去了意义。

本文介绍了一种新颖的潜变量模型，旨在缓解或规避上述局限性。我们的方法基于双重重构视角，其核心观点是，在源句子重构的示例中，将翻译后的目标句子视为源句子（观测变量）对应的潜变量 [14, 15]。该视角将源到目标和目标到源的翻译分别解释为推断和生成，从而构成对源句子的重构。类似地，目标句子的重构过程遵循相似框架，但视角相反。在观测变量和潜变量之间，我们引入一个额外的中间潜变量，从而建立双层次潜变量模型。该中间潜变量的一个关键特征是，它在两种语言的重构过程中共享。图1 展示了我们提出的双层次潜变量模型与传统方法之间的区别。传统方法包含使用不同潜变量的独立翻译过程，而我们的方法则共享一个中间潜变量，该变量由两个重构过程共同推断并用于生成。

共享的中间潜变量有助于同时生成源句子和目标句子，从而促进从源句子和目标句子中提取与语言无关的信息。此外，受双重重构视角的启发，我们利用两个编码器，在重构过程中分别充当推断模型和生成模型，共同推断共享中间潜变量的后验分布。这种方法类似于阶梯 VAE（LVAE）中采用的先进推断技术（阶梯推断），其中推断模型和生成模型协同推断后验分布 [16]。因此，我们将提出的模型称为 $LadderNMT$。值得注意的是，这一特性使我们提出的潜变量建模无需辅助后验模型，从而节省了参数量。更重要的是，这

阶梯推断机制通过一个简单的函数显式纳入源信息和目标信息，显著缓解了 OSPC问题。

在我们的实验结果中，我们证明 LadderNMT 的翻译性能优于以往的方法，尤其是优于最先进的非自回归翻译模型 FullyNAT [7], ，同时参数量平均减少约 30%。此外，通过对中间潜在空间进行定性分析，我们展示了 LadderNMT 在提取语言无关信息方面具有良好的正则化效果。最后，实验研究揭示了传统方法中存在 OSPC 问题，而 LadderNMT 能够有效规避这一问题。

2 相关工作

2.1 变分神经机器翻译

神经机器翻译的任务是将源句子映射到目标句子，这可以理解为一种多对多映射。例如，一个源句子可能对应多个潜在的目标句子，具体取决于风格偏好和预期交际目的等因素。然而，传统的解码策略仅依赖于源句子的确定性编码表示，可能难以有效处理这种细微复杂的映射。为应对这一挑战，变分神经机器翻译（VNMT）应运而生[17, 18]。VNMT 通过对潜变量 Z 进行边缘化，估计给定源句子的目标句子的似然，记作 $p(Y|X)$，如以下公式所示：

$$p(Y|X) = \int_Z p(Y, Z|X)dZ = \int_Z p(Y|Z, X)p(Z|X)dZ,$$

其中， $X = \{x_1, x_2, ..., x_{T_x}\}$ 是由 T_x 个源词 $x \in V_x$组成的源句子， V_x 是源语言的词汇集。同理， $Y = \{y_1, y_2, ..., y_{T_y}\}$是包含 T_y 个目标词 $y \in V_y$的目标句子。 $Z = \{z_1, z_2, ..., z_{T_z}\}$ 是由 T_z 个 $z \in \mathbb{R}^{d_z}$向量构成的潜变量集合。值得注意的是，对 Z 进行采样所具有的随机性，有助于表示多模态翻译。为避免从计算开销巨大的真实后验分布中采样，VNMT采用变分后验分布 $q(Z|X, Y)$来近似 $p(Z|X, Y)$。最后，目标函数由ELBO函数定义如下：

$$\mathcal{J}_{vnmt}(X, Y) = \mathbb{E}_{q(Z|X,Y)}[\log p(Y|Z, X)] - \alpha D_{KL}[q(Z|X, Y) \parallel p(Z|X)], \quad (1)$$

其中， D_{KL} 表示 Kullback-Leibler散度（KL）。似然分布和先验分布 $p(Y|Z, X)$ 与 $p(Z|X)$ 分别由神经机器翻译的编码器模型和解码器模型参数化。近似后验分布 $q(Z|X, Y)$ 通常由另一个编码器-解码器模型参数化。在测试过程中，对 Z 进行采样。

来自先验分布 $p(Z|X)$，由于目标句子不可用，因此需要最小化 KL 散度项，以使先验分布与后验分布保持一致。α 是用于调节 KL 散度影响的系数。

本文探讨了这一框架中的若干挑战。首先，ELBO 方程中的潜变量仅用于计算目标句子似然，这可能导致模型在训练期间捕获目标语言特有的信息，而这些信息在测试时既不可取也不可用。其次，辅助后验模型引入了大量参数，通常采用 Transformer 架构 [2]。最后，高复杂度模型与两个目标项重要性不平衡相结合，可能导致源句子与目标句子之间出现单侧信息提取，这一问题称为单侧后验坍塌（OSPC）问题。例如，如果似然项的重要性更高，那么潜变量可能会保留目标句子中的信息，同时忽略源句子中的信息。另一方面，如果 KL 散度项的重要性更高，后验分布可能会完全依赖源句子，从而获得使 KL 散度最小化的最优解：$q(Z|X, Y) = p(Z|X)$。本文还研究了传统潜变量模型中是否存在 OSPC 问题。

自 VNMT 出现以来，已有多项进展 [18–20]。值得注意的是，一些早期研究提出在源句和目标句之间共享潜在空间，并利用这一共享潜变量来估计神经机器翻译和语言模型（LM）的似然，分别记为 $p(Y|Z, X)$ 和 $p(X|Z)$[19]。特别是，MGNMT [20], 是一种自回归模型，它提出在源端和目标端并行执行神经机器翻译和语言模型任务。这些方法表明，利用潜变量同时生成 X 和Y 能够封装语言无关信息，从而提升翻译性能。然而，尽管取得了这些进展，仍然存在依赖辅助后验模型的问题。此外，先前的工作将源句和目标句都视为观测变量，并认为它们同时与潜变量对齐，因此采用标准高斯分布作为先验分布，而不以输入句子为条件：$p(Z)$，而非式1中的 $p(Z|X)$。这会使 KL 散度项的正则化缺乏信息，并且在测试期间可能需要许多迭代步骤，才能从随机噪声中逐步优化潜变量。相比之下，我们提出的双层次潜变量模型建立在 LAVE 的梯形推断方法之上，无需辅助后验模型，并允许采用以输入句子为条件的先验分布。因此，该模型减少了参数量，并有助于在测试期间从条件先验分布中采样得到质量更高的初始潜变量。

2.2 非自回归神经机器翻译（非自回归翻译）

利用潜变量建模，神经机器翻译模型的自回归解码过程遵循以下公式：

$$p(Y|Z,X)=\prod_{i=1}^{T_y}p(y_i|Z,X,Y_{<i}).$$

由于右侧乘积项中的条件概率取决于此前生成的目标词，翻译速度会随着句子长度的增加而成比例下降。

为提升速度，人们引入了非自回归解码 [3]。每个目标词的条件概率与其前面的目标词相互独立：

$$p(Y|Z,X)=\prod_{i=1}^{T_y}p(y_i|Z,X).$$

因此，解码过程可以完全并行执行，与自回归方法相比，显著提升了速度。自非自回归翻译（非自回归翻译）问世以来，便受到广泛关注，并由此催生了大量方法 [4–11]。

然而，由于目标词之间缺乏依赖关系，翻译速度的提升往往以牺牲翻译质量为代价，从而导致“多模态问题” [3]。VNMT框架中的潜变量建模已成为解决这一挑战的有效方案，因为它能够隐式地为整个目标序列解码表示统一模式。采用VNMT框架潜变量建模的非自回归翻译方法，例如 LaNMT [6] 和 FullyNAT[7], ，其结构大多遵循第2.1节开头所述的传统框架。基于潜变量设计，FullyNAT采用连接主义时序分类（CTC）损失 [5] ，并在翻译质量和翻译速度方面均达到最先进水平（SOTA）。

另一方面，还存在基于非潜变量的建模。DA-transformer[9, 10] 被提出用于预测目标词元之间的有向无环图依赖关系，而 PCFG-NAT [11] 则将图依赖关系推广到涵盖非相邻和双向结构。尽管这些近期研究也被认为达到了最先进的性能，但尚未与 FullyNAT 进行直接比较。由于这些方法依赖分词后的 BLEU 分数 [21], ，而该分数取决于特定的预处理方法，因此直接比较报告的 BLEU 分数并不恰当。根据这些方法相较于自回归对应基线所报告的性能提升，我们谨慎地认为，FullyNAT 在翻译质量和速度方面均优于基于非潜变量的方法。本研究将一种新颖的潜变量模型应用于现有的优秀潜变量建模方法 LaNMT 和 FullyNAT，从而阐明了我们所提出方法的优势。

3 所提方法

本节介绍我们所提出的新型潜变量模型的结构框架和目标，详见 3.1 节。随后，我们将在 3.2 节说明各个分布的设计，包括似然、先验和后验，并介绍所提出的 LadderNMT 模型中的编码器和解码器参数。值得注意的是，我们的后验估计策略基于 LVAE 所采用的阶梯推断技术。我们将详细阐述这一特定设计在缓解 OSPC问题方面的优势。最后，我们将在 3.3 节介绍一种方法，在两个重建过程中部分共享中间潜变量。我们将重点说明，与完全共享相比，部分共享可能更有效，尤其是在两种语言需要具有一定程度不同的潜在空间时。

3.1 双层次潜变量模型

如引言中所述，我们所提方法基于双重重构的概念，该概念将翻译句子视为输入句子的潜变量，并同时进行两个重构过程。此外，我们引入了一个由输入句子和翻译句子共享的中间潜变量，从而建立了一个双层次潜变量模型。本节将推导我们方法的目标函数，首先从源端重构过程开始。由于我们关注的是翻译任务，因此将这一推导扩展到监督学习场景，在该场景中，数据集提供翻译后的目标句子。换言之，源句子的翻译目标句子这一事件是经过采样后固定的。最后，我们给出最终目标函数，其中包含源端重构目标和目标端重构目标。

源端重构过程的目标是最大化 $\log p(X)$。按照包含两个潜变量（Z， Y）的变分推断 [22]的常规推导，其中 Y表示翻译后的目标句子，目标可推导为证据下界形式：

$$
\begin{aligned}
\log p(X) &\geq \int_Z \sum_{\hat{Y}} q(Z,\hat{Y}|X) \log \left(p(X|Z,\hat{Y}) \frac{p(Z,\hat{Y})}{q(Z,\hat{Y}|X)} \right) dZ \\
&= \int_Z \sum_{\hat{Y}} q(Z|\hat{Y},X) q(\hat{Y}|X) \log \left(p(X|Z,\hat{Y}) \frac{p(Z|\hat{Y}) p(\hat{Y})}{q(Z|\hat{Y},X) q(\hat{Y}|X)} \right) dZ \\
&= \sum_{\hat{Y}} q(\hat{Y}|X) \int_Z q(Z|\hat{Y},X) \log p(X|Z,\hat{Y}) dZ \\
&\quad + \sum_{\hat{Y}} q(\hat{Y}|X) \int_Z q(Z|\hat{Y},X) \log \frac{p(Z|\hat{Y})}{q(Z|\hat{Y},X)} dZ \\
&\quad + \sum_{\hat{Y}} q(\hat{Y}|X) \log \frac{p(\hat{Y})}{q(\hat{Y}|X)} \int_Z q(Z|\hat{Y},X) dZ.
\end{aligned}
$$

目标公式的第一项表示由层次潜变量边缘化后的观测对数似然。第二项和第三项分别表示中间潜变量和翻译句子的 KL 散度。根据推导出的这一目标，将 Y （翻译句子）替换为 Y （真实目标句子），即可得到监督学习情形。由于 Y 是 X的对齐潜变量，我们消去 Y的边缘化过程。基于蒙特卡洛近似以及数据集中不存在重复句子的假设，我们假设 $p(Y)$和 $p(X)$均为均匀分布，对每个可能结果赋予 $1/N$概率质量，其中 N 是数据样本数。此外，基于真实句对 X 和Y之间的翻译是一一映射的假设，我们假设 $q(Y|X)$和 $q(X|Y)$均为 1。最后，监督学习中的目标可近似为:

$$\begin{aligned}\mathcal{J}_X(X,Y) &\approx \int_Z q(Z|Y,X)\log p(X|Z,Y)dZ + \int_Z q(Z|Y,X)\log\frac{p(Z|Y)}{q(Z|Y,X)}dZ + \log\frac{1}{N}\\ &\approx \int_Z q(Z|Y,X)\log p(X|Z,Y)dZ + \int_Z q(Z|Y,X)\log\frac{p(Z|Y)}{q(Z|Y,X)}dZ\\ &= \mathbb{E}_{q(Z|Y,X)}[\log p(X|Z,Y)] - D_{KL}[q(Z|Y,X)\parallel p(Z|Y)].\end{aligned}$$

为简化起见，我们进行了第二次近似，因为常数 $\log\ 1/N$ 在优化过程中不会影响最优解。类似地，目标句子重建的目标函数推导如下:

$$\mathcal{J}_Y(X,Y) = \mathbb{E}_{q(Z|X,Y)}[\log p(Y|Z,X)] - D_{KL}[q(Z|X,Y)\parallel p(Z|X)].$$

最后，双层次潜变量模型的总目标函数定义如下:

$$\begin{aligned}\mathcal{J}(X,Y) &= \mathcal{J}_X(X,Y) + \mathcal{J}_Y(X,Y)\\ &= \mathbb{E}_{q(Z|X,Y)}[\log p(X|Z,Y) + \log p(Y|Z,X)]\\ &\quad - \alpha\left(D_{KL}[q(Z|X,Y)\parallel p(Z|X)] + D_{KL}[q(Z|X,Y)\parallel p(Z|Y)]\right). \qquad (2)\end{aligned}$$

值得注意的是，由于共享潜在空间，中间潜变量 Z 是从近似后验分布 $q(Z|X,\ Y)$ 中采样得到的，并用于似然和 KL 散度。

3.2 应用于非自回归翻译

我们将双层次潜变量模型应用于非自回归翻译任务。给定两个编码器-解码器模型，$\theta = \{\theta^e, \theta^d, \theta^{pri}\}$ 和 $\phi = \{\phi^e, \phi^d, \phi^{pri}\}$，它们分别用于源语言到目标语言和目标语言到源语言的翻译。 θ^e 和 θ^d (或 ϕ^e 和 ϕ^d)分别表示编码器和解码器模型的参数。通常，这些模型采用 Transformer 架构 [2]，但解码器自注意力层的因果掩码设计除外，以便一次性翻译整个句子。 θ^{pri} (或 ϕ^{pri})是线性层的权重参数，用于根据编码器计算出的表示估计先验高斯分布的均值和方差

。值得注意的是，后验估计没有对应的参数，后文将对此作进一步说明。对于先前的工作 LaNMT [6], ，还包含一组额外参数，用作长度预测器，根据中间潜变量预测翻译句子的长度。另一方面，FullyNAT [7] 不使用长度预测器，而是直接预测 U 个词元（数量为输入句子长度的倍数），然后直接删除连续重复的词元。我们的 LadderNMT 采用 LaNMT 或 FullyNAT 的长度预测方法。

3.2.1 先验分布设计

源端重构过程的先验分布记为 $p(Z|X)$，由一个 Transformer 编码器（包括词嵌入层 [23]）和由 θ^e 与 θ^{pri} 参数化的线性层建模。首先，Transformer 编码器将源句子转换为确定性隐藏表示，表示为 $TE(X;\theta^e) = H^x = \{h_1^x, h_2^x, ..., h_{T_x}^x\}$，其中 TE 表示 Transformer 编码器函数。随后，对每个隐藏表示进行线性变换，以估计高斯先验分布的均值和方差，具体如下：

$$
\begin{aligned}
p(z_i|X;\theta^e,\theta^{pri}) &= \mathcal{N}(z_i|\mu_i^x,\sigma_i^x), \quad 1 \le i \le T_z \\
\mu_i^x &= LT(linear(H^x;\theta_\mu^{pri}), T_z)_i \in \mathbb{R}^{d_z}, \\
\sigma_i^x &= LT(softplus(linear(H^x;\theta_\sigma^{pri}), T_z)_i \in \mathbb{R}^{d_z},
\end{aligned}
$$

其中 $softplus(a) = \log(1+e^a)$。这里，$LT(\cdot, L)$ 表示长度变换函数，它会将输入向量序列的长度调整为指定长度 L。由于两种语言的重构共享中间潜在空间，因此潜变量需要统一的序列长度 T_z。我们采用单调位置注意力机制，将输入向量以平滑扩散的权重融入相邻位置，详情见先前的工作 [6]。对于将 LadderNMT 应用于 LaNMT 模型的情况，T_z 设为预定义值。相反，对于将其应用于 FullyNAT 模型的情况，长度会动态变换为 $T_z = U(T_x + T_y)/2$。

同样，目标端重构过程的先验分布 $p(Z|Y)$ 也按照类似的步骤建立，并相应地替换输入句子 Y 和参数 ϕ^e 和 ϕ^{pri}。

3.2.2 后验分布设计：阶梯推断

如引言中简要提到的，阶梯推断需要同时利用推断模型和生成模型来估计后验分布。在我们提出的双层次潜变量模型中，源到目标 θ 和目标到源 ϕ 模型同时承担推断和生成的角色。因此，μ^x 和 σ^x 由 θ 计算，被视为由推断模型估计的参数；而从源重构的角度来看，μ^y 和 σ^y 由 ϕ 计算，被视为由生成模型估计的参数。随后，阶梯

根据以下公式进行推断，以计算后验分布：

$$
\begin{aligned}
q(z_i|X,Y;\theta^e,\theta^{pri},\phi^e,\phi^{pri}) &= \mathcal{N}(z_i|\mu_i^{xy},\sigma_i^{xy}),\\
\mu_i^{xy} &= \frac{\mu_i^x \odot (\sigma_i^x)^{-1} + \mu_i^y \odot (\sigma_i^y)^{-1}}{(\sigma_i^x)^{-1} + (\sigma_i^y)^{-1}},\\
\sigma_i^{xy} &= \frac{1}{(\sigma_i^x)^{-1} + (\sigma_i^y)^{-1}},
\end{aligned}
$$

其中，$\odot$ 表示逐元素乘法运算。

当估计得到的后验分布趋于独立于某个条件项时，就会出现 OSPC问题。如第2.1节所举示例所示，当后验估计器足够复杂，能够忽略输入信息的一侧，且目标项的重要性不平衡时，就会出现这种情况。对于前一种因素，阶梯推断具有一定的抵抗能力，因为它通过简单的动态机制纳入了两种先验分布的均值和方差。除非出现极端情况，例如先验估计器赋予零均值或无限大的方差，否则很难忽略单侧信息；这与以往基于Transformer的后验模型形成对比。

此外，我们认为，阶梯推断的结构能够防止因 KL 散度的重要性过度偏大而导致的 OSPC问题。为证实这一观点，我们推导了一个 KL 散度项的解析解，$D_{KL}[q(Z|X,Y) \parallel p(Z|Y)]$。鉴于后验分布和先验分布均服从高斯族，该 KL 散度项可推导如下：

$$
D_{KL}[q(Z|X,Y) \parallel p(Z|Y)] = \log\frac{\sigma^y}{\sigma^{xy}} + \frac{(\sigma^{xy})^2 + (\mu^{xy}-\mu^y)^2}{2(\sigma^y)^2} - \frac{1}{2}.
$$

KL 散度关于 μ^x 的一阶和二阶导数推导如下：

$$
\frac{\partial D_{KL}}{\partial \mu^x} = (\mu^x - \mu^y)\frac{1}{(\sigma^x+\sigma^y)^2}, \quad \frac{\partial^2 D_{KL}}{(\partial \mu^x)^2} = \frac{1}{(\sigma^x+\sigma^y)^2} \geq 0.
$$

由此可推知，当μ^x 等于 μ^y时，KL 散度达到最优解。因此，KL 散度项的重要性较高，并不意味着模型必须忽略输入信息的一侧；相反，它会对两个不同的先验估计器进行正则化，使其推断出相同的均值。

3.2.3 似然分布设计

源端重构过程的似然分布记为 $p(X|Z,\ Y)$，采用 Transformer 解码器建模，其参数为 ϕ^d；该解码器不包含因果掩码和词嵌入层，但包含 logit线性层。该解码器以编码后的隐藏表示 H^y 和训练期间从后验分布 $q(Z|X,\ Y)$中采样得到的潜变量 Z 作为输入。在测试期间，从

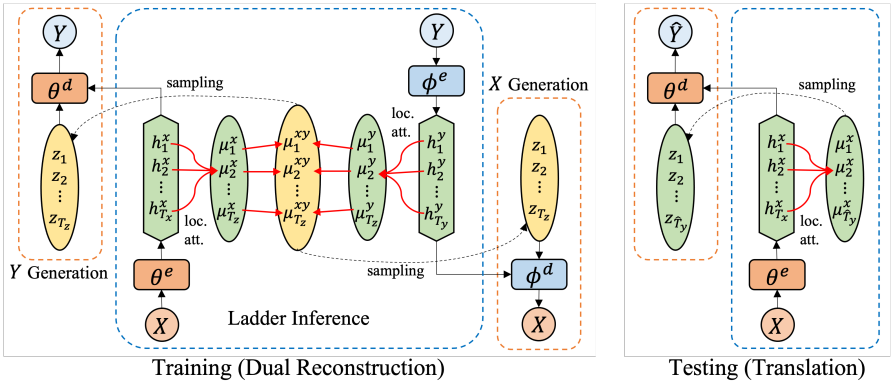


图2 基于我们提出的 LadderNMT 方法进行训练（左侧，双重重构任务）和测试（右侧，翻译任务）的示意图。测试部分仅展示源语言到目标语言翻译任务；目标语言到源语言翻译任务则以对称方式，使用相反的模型和输入 Y完成。 μ 项分别表示各个先验和后验分布。为简洁起见，我们省略了先验和后验分布中的方差参数元素 σ 。“loc. att.”表示将总长度转换的单调位置注意力机制。

先验分布 $p(Z|Y)$中采样。为实现反向传播，采样过程采用重参数化技巧 [13],，其定义为 $z_i = \mu_i^{xy} + \boldsymbol{\epsilon}\sigma_i^{xy}$ 其中 $\boldsymbol{\epsilon} \sim \mathcal{N}(000, \boldsymbol{I})$。随后，似然分布计算如下:

$$p(x_i|Z, Y; \phi^d) = softmax(O_i^x),$$
$$O_i^x = TD(Z, H^y; \theta^d)_i \in \mathbb{R}^{|V|},$$

其中， TD 表示 Transformer 解码器函数。类似地，目标端重构过程的 $p(Y|Z, X)$ 使用另一个以 θ^d 为参数、输入为 Z 和 H^x 的 Transformer 解码器计算。需要注意的是，由于采用了共享潜在空间，源端和目标端重构过程使用相同的 Z 。

LadderNMT 的整个训练（监督学习）和测试过程如图所示：2。训练期间，首先，两个编码器计算编码后的隐藏表示， H^x 和 H^y。其次，两个先验估计器分别估计先验分布。第三，梯式推理过程根据这些先验分布计算后验分布。第四，在对中间潜变量进行采样后，两个解码器生成 X 和 Y。最后，双层次潜变量模型的总目标函数（如公式 2所示）被计算出来，并对整个模型进行联合优化。测试期间，我们遵循与训练期间类似的过程，但中间潜变量从先验分布中抽取。然而，在翻译的第一次迭代之后，在迭代细化设置下，我们可以遵循梯式推理过程，根据第一次翻译结果给出的后验分布对中间潜变量进行采样 [6]。

3.3 部分潜在空间共享

我们提出了一种变体，不再像 LadderNMT 那样共享整个潜在空间，而是部分共享中间潜在空间。这种方法将部分维度分配给每种语言信息，使其能够专注于各自的语言信息。共享维度与非共享维度的比例由预先设定的超参数 $\rho \in [0, 1]$ 控制。共享维度的数量计算为 $d_z^{srd} = round(d_z \times \rho)$，其中 d_z 是中间潜在向量的维度。因此，非共享维度的数量为 $d_z - d_z^{srd}$。

当不同语言的中间潜变量真实流形存在显著差异时，部分共享可能具有优势。这种差异可能源于地理、文化或历史方面的重大差别。例如，来自努纳维克的因纽特语方言（北极魁北克）中有许多表达“雪”的说法 [24]，而赤道地区的语言可能只有少数几个表示“雪”的词。这表明，这些语言的潜在空间可能在某些特定方面有所不同，例如与“雪”相关的词汇。在这些情况下，纳入非共享维度有助于保留这些独特的语言特征。

4 非自回归神经机器翻译结果

4.1 数据描述

我们的实验在三个翻译任务上进行：WMT14 英语-德语（En-De，390 万对）、WMT18 英语-芬兰语（En-Fi，290 万对）和 WMT18 英语-土耳其语（En-Tr，20 万对）[25, 26]。我们采用了 Fairseq 工具包 [27] 中使用的相同预处理和分词方法（包括子词字节对编码）。对于词汇集，我们分别从各自的训练数据集中选取出现频率最高的 32K、10K 和 10K 个子词。验证时，En-De、En-Fi 和 En-Tr 任务分别使用 Newstest13（3K 对）、Newstest16（3K 对）和 Newsdev16（1K 对）。测试时，我们分别使用 Newstest14（3K 对）、Newstest18（3K 对）和 Newstest18（3K 对）。

序列级知识蒸馏（KD） [12] 已用于缓解非自回归翻译训练中的多模态问题。该方法使用预训练的自回归翻译（AT）模型重新生成成对句子，从而减少一对多映射的示例。双向知识蒸馏方法进一步利用预训练的源到目标和目标到源 AT 模型生成句子，从原始数据集 [28]生成两个成对的知识蒸馏数据集。在所有实验中，我们都采用了双向知识蒸馏方法，因为双重重构视角本身就涉及两个方向的翻译，因此需要两个知识蒸馏数据集。我们使用基于基础 Transformer 架构的预训练 AT 模型 [2] 来重新生成这些知识蒸馏数据集。

4.2 模型与训练

如第 2.2 节所述，我们将 LadderNMT 模型应用于 LaNMT 和 FullyNAT 的架构（+变分自编码器模型 [7]）。我们基于自己的 Transformer 模型实现，重新实现了 LaNMT 和 FullyNAT，并使用了它们公开提供的源代码[1][2]。对于英德和英芬实验，我们采用了 [2]中详述的基础 Transformer 架构。对于英土实验，我们采用了小型 Transformer 架构；除前馈层维度为 1024、多头注意力层有 4 个注意力头、学习率为 0.0005、dropout 率为 0.3 外，其余配置均与基础架构相同。英德、英芬和英土任务中每个小批量的词元数量分别设为 8K、4K 和 4K。所有实验均采用逆平方根学习率调度器，并集成最新的 10 个检查点 [27]。根据验证结果保存最佳检查点；如果连续 50 个耐心步数内验证结果均未超过此前的最佳结果，则实施早停。

除上述基本配置外，对于与非自回归翻译方法相关的 LaNMT 和 FullyNAT，我们遵循了相同的配置 [6, 7], ，例如辅助后验模型配置、长度预测方法、中间潜在空间维度、 α的退火计划，以及潜变量的上采样比率（针对 FullyNAT）。值得注意的是，我们将 FullyNAT 基线的默认 α 设置为 0.5，这一数值根据我们的内部消融研究被证明是最优的。与 FullyNAT 基线配置存在显著差异的一项是每个小批量中的词元数量：我们对英德任务使用了 8K，而他们使用的是 128K。由于计算和内存资源的限制，这一调整是必要的。需要注意的是，FullyNAT 的上采样操作使解码器部分的计算和内存开销增加了三倍，而 CTC损失在通过动态规划算法对可能路径进行边缘化时，也会增加此类开销。因此，与原论文相比，我们对 FullyNAT 基线的重新实现导致 BLEU分数下降了约 1.0 分。然而，为了进行公平比较，我们将 LadderNMT 应用于 FullyNAT 架构时采用了一致的配置。

关于 LadderNMT 应用的配置，我们通常遵循基线配置。然而，通过实证研究，我们确定了中间潜在空间维度和 KL 散度系数（d_z/α）的最优配置：英德为（16/1.5），英芬为（32/1.5），英土为（32/1.0）。我们还改变了部分共享比例 ρ 并选取了最佳值：英德为 0.25，英芬为 0.25，英土为 1.00。值得注意的是，我们的 LadderNMT 模型在 d_z 大于基线最优配置的值 $d_z = 8$下取得了最佳性能，这表明，与基线相比，LadderNMT 的共享中间潜在空间能够捕获更多相互独立的信息。此外，我们发现， α 超过基线最优值的值 $\alpha = 1.0$ （FullyNAT 为 0.5）进一步支持了我们关于 KL 散度正则化收益的论点，具体讨论见第 3.2.2节。

---

[1]https://github.com/zomux/lanmt [2]
https://github.com/shawnkx/Fully-NAT

所有英德实验均在两块 NVIDIA RTX4090 图形处理器上进行，平均训练时间约为 240 小时。对于英芬和英土任务，我们使用了两块 NVIDIA GTX1080Ti 图形处理器，平均训练时间约为 220 小时。

4.3 解码与评估

基于 LaNMT 架构的模型，其解码过程包含三个迭代细化步骤。对于基于 FullyNAT 架构的模型，则只需删除连续重复的词元。对于自回归 Transformer 基线模型，我们使用束搜索算法，束宽为 5。

为评估翻译质量，对于基于 LaNMT 架构的模型，我们采用不区分大小写的去标记化 BLEU 指标 [21] ，遵循原论文使用的评估指标[3]。对于基于 FullyNAT 架构的模型，我们采用原论文使用的分词 BLEU 指标。此外，我们报告了使用单张 GTX1080Ti 图形处理器在每个测试集上测得的平均翻译速度（毫秒）。最后，我们比较了各模型的参数总量。

表 1 基于 LaNMT 架构的模型的实验结果。 P 表示参数总量，单位为百万（M）。每个测试集所在列中"/"左侧和右侧的数字分别表示英语到$language$ 以及$language$到英语的翻译质量（BLEU分数）。翻译速度（Time）以毫秒（ms）为单位。术语"+层"表示 LadderNMT 模型增加了层数，以使其参数量与 LaNMT 基线模型相近：具体而言，我们分别为英德、英芬和英土任务增加了 5、3 和 4 层。

	WMT14 英德			WMT18 英芬			WMT18 英土		
模型	P	News14	Time	P	News18	Time	P	News18	Time
Transformer	154	26.5/32.0	162	109	14.1/20.0	191	64	16.0/19.5	196
LaNMT	232	23.8/30.2	22	164	11.6/17.7	13	98	12.5/16.3	14
使用 LadderNMT	155	24.9/30.4	28	109	12.3/17.7	17	66	12.6/16.4	15
+ 层	229	25.5/31.2	38	154	12.4/17.9	24	95	12.9/16.9	23

4.4 翻译结果

表1 展示了基于 LaNMT 架构的模型实验结果。与 LaNMT 相比，我们将 LadderNMT 应用于 LaNMT 基线（"使用 LadderNMT"），显著减少了参数量，同时取得了相当或更高的（英德任务）BLEU分数。尽管与 LaNMT 相比，翻译速度略有下降，但仍显著快于自回归 Transformer（自回归翻译）基线。此外，当我们增加 LadderNMT 的层数，使参数量与 LaNMT 相匹配时，LadderNMT+层数在数据集规模相对较大的 WMT14 英德任务上显著优于 LadderNMT

[3]尽管他们的论文将 SacreBLEU [29] 描述为评估指标，但其源代码似乎使用的是采用"intl"分词器、且不区分大小写的去标记化 BLEU。

表 2 基于 FullyNAT 架构的模型实验结果。每个测试集列中"/"左右两侧的数字分别表示英语到 $German$ 以及 $German$到英语的翻译质量（BLEU分数）。

	WMT14 英德		
模型	参数量(M)	News14	时间(ms)
Transformer	153	27.6/33.6	162
FullyNAT	166	26.2/30.4	6
使用 LadderNMT	121	26.9/31.5	7

的数据集规模相对较大。我们认为，所提出的方法采用了高效的架构，参数量较少，因此可以扩展模型的其他部分，以提升性能。

表2展示了基于最先进的非自回归翻译模型 FullyNAT 架构的模型在 WMT14 英德任务上的实验结果。值得注意的是，我们将 LadderNMT 调整为 FullyNAT 架构（"使用 LadderNMT"）后，英译德和德译英任务的 BLEU分数均有所提高，同时参数总量减少了约 27%。此外，与 FullyNAT 基线相比，这种调整并未显著降低翻译速度。这些结果表明，我们的方法可以减少现有非自回归翻译模型的参数量，同时保持（或提升）翻译质量和速度。

5 共享中间潜在空间的定性分析

为了了解我们提出的方法对共享中间潜在空间的影响，我们对该空间进行了定性分析。下文为简洁起见，将中间潜变量简称为潜变量。在第5.1 节和第5.2节所述的分析中，我们使用了在英德任务上预训练的 LaNMT 和 LadderNMT（应用于 LaNMT）模型。使用 Newstest14 收集潜变量。具体而言，潜变量 Z 从先验分布中采样得到，$p(Z|X)$ 和 $p(Z|Y)$。换言之，LadderNMT 收集的潜变量由两个编码器和先验估计器分别独立计算，其参数为 (θ^e, θ^{pri}) 和 (ϕ^e, ϕ^{pri})，以反映测试场景。对于第5.3节所述的 OSPC 分析，我们收集了 Z 从后验分布中采样得到的$q(Z|X,Y)$。

5.1 二维可视化

首先，我们使用 t-SNE 将收集到的 LadderNMT 和 LaNMT 潜变量可视化为二维空间中的点 [30] ，并使用 UMAP [31],进行可视化，如图所示。3。红点表示英语的潜变量，蓝点表示德语的潜变量。LaNMT 的潜变量按语言明显区分。需要注意的是，我们认为 LaNMT 代表的是一种可比基线，即分别训练的两个

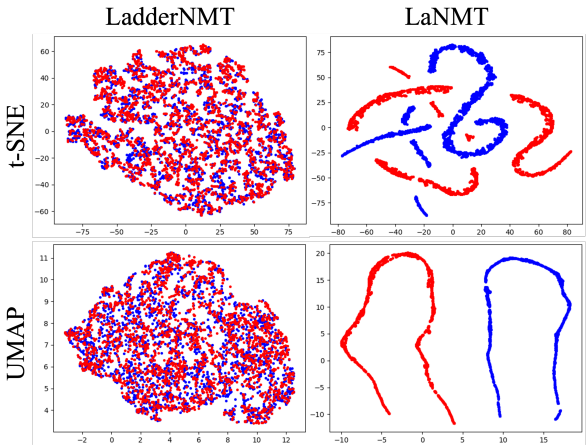


图 3 通过 t-SNE 和 UMAP 将 LadderNMT 和 LaNMT 的潜变量映射到二维空间。红色和蓝色的点分别表示来自英语和德语句子的潜变量。

潜在空间，而不是说这些潜变量包含对翻译任务不利的语言特定信息。相反，LadderNMT 的潜变量无法按语言区分，这表明潜在空间共享具有正则化效果，有助于提取语言无关信息，而非语言特定信息。

5.2 典型相关分析（CCA）

虽然第 5.1 节表明 LadderNMT 提取了与语言无关的信息，但仍需验证配对句子的潜变量是否如第3.2.2 节（LadderNMT 中 KL 散度的最优解推导）所述紧密映射。本节将考察潜在空间中配对句子的相似性。我们采用典型相关分析（CCA） [32] ，并测量 CCA 预测得分，以评估潜在空间中配对句子的相似性。CCA 方法包含两个可训练的线性变换层，分别作用于两个配对的输入模态；训练时，这两个层会在低维空间（线性层的输出空间）中最大化配对数据之间的典型相关性。预测得分的公式如下：$\left(1-\frac{(\hat{}Z_X-\hat{}Z_Y)^2}{(\hat{}Z_X-m(\hat{}Z_X))^2}\right)$，其中 $m(\hat{}Z_X)$ 是所有 Z_X 的均值。这里， Z_X 和 Z_Y 是配对的英语和德语潜变量，其维度经过线性变换后降低。CCA 预测得分越高，表明配对句子的潜变量映射得越紧密。

CCA 参数使用从验证集（Newstest13）中收集的潜变量进行训练，随后应用于从 Newstest14 中收集的潜变量。LadderNMT 的得分为 **0.6851**，而 **LaNMT 的得分为 0.2836** 。结合第5.1节的分析，这表明 LadderNMT 不仅能提取语言无关信息，还能将成对句子紧密地映射到一起。我们认为，这是我们的 KL 散度所产生的正则化效果所致，其作用于

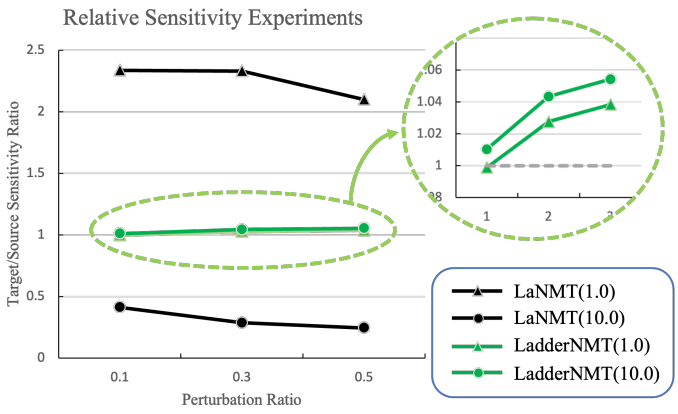


图 4 相对敏感性测试结果。模型标签旁括号中的数字表示 KL 系数。

目标函数（定义见式 2），该函数有效地对齐了两个编码器及其先验估计。

5.3 OSPC问题研究的相对敏感性

作为衡量 OSPC问题的一种方法，我们提出了一种新方法——“相对敏感性”测试，用于评估模型对添加到源句子或目标句子中的输入噪声的相对敏感程度。在这项测试中，我们交替（而非同时）随机改动源句子或目标句子中的若干单词，并计算从受扰动句子和原始句子推断出的潜变量之间的欧几里得距离。没有 OSPC问题的模型对源句子和目标句子输入中的扰动应表现出相近的敏感程度，因此相对敏感性比值接近 1。

为明确验证原始证据下界和我们的目标函数中是否存在 OSPC问题，方程 1 和 2中所示，我们使用两种不同的 KL 散度系数 α：1.0 和 10.0，训练了 LaNMT 基线模型和我们的 LadderNMT 模型。图 4展示了 LaNMT 和 LadderNMT 在这些不同 KL 散度系数下的相对敏感性。正如预期，LaNMT 的相对敏感性明显偏离 1.0 线，表明其后验模型忽略了某一侧的信息。相比之下，无论 KL 散度系数如何，LadderNMT 的相对敏感性始终接近 1.0。这一结果证实了我们在第 3.2.2节中的论断：LadderNMT 不易受到与 KL 散度项重要性相关的 OSPC问题的影响。

6 结论

我们提出了一种新颖的非自回归神经机器翻译潜变量建模方法，该方法基于对翻译任务的不同视角以及一种先进的层次化潜变量模型。与传统的潜变量建模不同，我们的方法通过对潜在空间进行正则化来提取信息丰富的特征。

从输入句子中提取信息，同时不增加后验估计模型的参数量。此外，这也说明后验估计对输入信息的各个部分仍然同样敏感。

我们的定量实验结果表明，与包括最先进模型在内的多个基线模型相比，所提方法能够实现更高的翻译质量。此外，对潜在空间的定性分析证实，LadderNMT能够推断出信息量更丰富、与语言无关的潜变量，并且能够抵抗后验估计对部分输入信息不敏感的问题。

未来的研究中，我们计划将所提方法应用于多语言神经机器翻译，因为我们认为，当涉及多种语言时，通过共享潜在空间产生的正则化效果可以得到增强。此外，我们的方法还适用于其他具有结构二元性的任务，例如文本到图像、图像到文本、文本到语音以及语音到文本生成任务。此外，我们的方法还可以应用于基于非潜变量的模型，例如 DA-Transformer，只需为其配备潜变量架构即可。

致谢。 本研究得到韩国教育部资助的韩国国家研究基金会基础科学研究项目（NRF-2022R1A2C1012633）的支持

声明

DongNyoneg Heo 和 Heeyoul Choi 报告称，他们获得了韩国国家研究基金会的资助。

参考文献

[1] Bahdanau, D.、Cho, K.、Bengio, Y.：通过联合学习对齐与翻译实现神经机器翻译。arXiv 预印本 arXiv:1409.0473（2014年）[2] Vaswani, A.、Shazeer, N.、Parmar, N.、Uszkoreit, J.、Jones, L.、Gomez, A.N.、Kaiser, Ł.、Polosukhin, I.：注意力就是你所需要的一切。NeurIPS **30** （2017年）[3] Gu, J.、Bradbury, J.、Xiong, C.、Li, V.O.、Socher, R.：非自回归神经机器翻译。arXiv 预印本 arXiv:1711.02281（2017年）[4] Lee, J.、Mansimov, E.、Cho, K.：通过迭代细化实现确定性的非自回归神经序列建模。arXiv 预印本 arXiv:1802.06901（2018年）[5] Libovický, J.、Helcl, J.：结合连接主义时序分类的端到端非自回归神经机器翻译。arXiv 预印本 arXiv:1811.04719（2018年）[6] Shu, R.、Lee, J.、Nakayama, H.、Cho, K.：使用 delta 后验进行确定性推理的潜变量非自回归神经机器翻译。载于：AAAI 会议论文集，第 34 卷，第 8846–8853 页（2020年）

[7] Gu, J.、Kong, X.：完全非自回归神经机器翻译：实用技巧。arXiv 预印本 arXiv:2012.15833（2020年）[8] Bao, Y.、Zhou, H.、Huang, S.、Wang, D.、Qian, L.、Dai, X.、Chen, J.、Li, L.：latent-glat：通过观察潜变量实现并行文本生成。载于：第 60 届 ACL 会议论文集，第 8398–8409 页（2022年）[9] Huang, F.、Zhou, H.、Liu, Y.、Li, H.、Huang, M.：用于非自回归机器翻译的有向无环 Transformer。载于：ICML，第 9410–9428 页（2022年）[10]Ma, Z.、Shao, C.、Gui, S.、Zhang, M.、Feng, Y.：非自回归机器翻译中有向无环图的模糊对齐。arXiv 预印本 arXiv:2303.06662（2023年）[11] Gui, S.、Shao, C.、Ma, Z.、Chen, Y.、Feng, Y. 等：基于概率上下文无关文法的非自回归机器翻译。NeurIPS**36** （2024年）[12]Kim, Y.、Rush, A.M.：序列级知识蒸馏。arXiv 预印本 arXiv:1606.07947（2016年）[13]Kingma, D.P.、Welling, M.：自动编码变分贝叶斯。arXiv 预印本 arXiv:1312.6114（2013年）[14] He, D.、Xia, Y.、Qin, T.、Wang, L.、Yu, N.、Liu, T.-Y.、Ma, W.-Y.：机器翻译的双重学习。NeurIPS **29**（2016年）[15] Heo, D.、Choi, H.：在反向翻译框架中对两个翻译模型进行端到端训练。arXiv 预印本 arXiv:2202.08465（2022年）[16]Sønderby, C.K.、Raiko, T.、Maaløe, L.、Sønderby, S.K.、Winther, O.：阶梯式变分自编码器。NeurIPS **29**（2016年）[17] Zhang, B.、Xiong, D.、Su, J.、Duan, H.、Zhang, M.：变分神经机器翻译。arXiv 预印本 arXiv:1605.07869（2016年）[18] Su, J.、Wu, S.、Xiong, D.、Lu, Y.、Han, X.、Zhang, B.：变分循环神经机器翻译。载于：AAAI 会议论文集，第 32 卷（2018年）[19] Shah, H.、Barber, D.：生成式神经机器翻译。NeurIPS **31** （2018年）[20] Zheng, Z.、Zhou, H.、Huang, S.、Li, L.、Dai, X.-Y.、Chen, J.：镜像生成式神经机器翻译。载于：ICLR（2019年）[21] Papineni, K.、Roukos, S.、Ward, T.、Zhu, W.-J.：BLEU：一种机器翻译自动评估方法。载于：第 40 届 ACL 会议论文集，第 311–318 页（ 2002 年）[22] Bishop, C.M.：《模式识别与机器学习》（信息科学

与统计学）。施普林格，柏林，海德堡（2006年）

[23]Mikolov, T., Chen, K., Corrado, G., Dean, J.：在向量空间中高效估计词表示。arXiv 预印本 arXiv:1301.3781（2013年）[24] 百科全书：因纽特语中表示雪和冰的词语。《加拿大百科全书》（2015年）[25] Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., 等：2014年统计机器翻译研讨会论文集。在：第9届统计机器翻译研讨会论文集，第12–58页（2014年）[26]Bojar, O., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Koehn, P., Monz, C.：2018年机器翻译会议（WMT18）成果。在：第3届机器翻译会议论文集，第2卷，第272–307页（2018年）[27] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., Auli, M.：fairseq：用于序列建模的快速、可扩展工具包。在：NAACL-HLT 2019：演示论文集（2019年）[28]Ding, L., Wang, L., Liu, X., Wong, D.F., Tao, D., Tu, Z.：重振低频词：充分利用非自回归翻译中的平行语料。arXiv 预印本 arXiv:2106.00903（2021年）[29]Post, M.：呼吁清晰报告 BLEU 分数。arXiv 预印本 arXiv:1804.08771（2018年）[30] Maaten, L., Hinton, G.：使用 t-SNE 可视化数据。《机器学习研究杂志》**9(11)**（2008年）[31] McInnes, L., Healy, J., Melville, J.：UMAP：用于降维的统一流形逼近与投影。arXiv 预印本 arXiv:1802.03426（2018年）[32] Hotelling, H.：两组变量之间的关系。《生物计量学》**28**(3/4)，321–377（1936年）。访问日期：2023年1月11日