Imperial College London

伦敦帝国学院计算系

arXiv:2603.00888v1 [cs.LG] 1 Mar 2026

深度序列模型中的概率学习与生成

Wenlong Chen

2025年12月18日

本学位论文为申请哲学博士学位而提交

原创性声明

我在此声明，本学位论文中的工作均为本人独立完成。他人的工作已得到适当引用。完整的参考文献列表见书目。

版权

本学位论文的版权归作者所有，其内容依据知识共享署名—非商业性使用—相同方式共享 4.0 国际许可协议（CC BY-NC-SA 4.0）提供。您可以通过任何媒介或格式复制和再分发本材料，也可以对本材料进行混编、转换或基于本材料创作。在此过程中，您必须适当注明作者，提供许可协议链接，并说明是否进行了任何修改。如果您对本材料进行混编、转换或基于本材料创作，则必须按照相同的许可协议再分发您的贡献。您不得将本材料用于商业用途。

对于上述许可未涵盖的本作品用途，请向版权所有者申请许可。

致谢

首先，我要感谢我的导师 Yingzhen Li，在我攻读博士学位期间始终给予我坚定不移的支持。无论从哪方面来说，Yingzhen 都是我所能想象到的最好的导师。她不仅对研究充满热情，而且知识渊博、耐心谦逊，与她共事也十分愉快。我很感激 Yingzhen 给予我自由和鼓励，让我得以追求自己的想法，探索不同的研究方向。她深刻的见解和坦诚的反馈总能启发我深入思考，帮助我成长为更出色的研究者。多亏了 Yingzhen，这四年成为我人生中最难忘的经历之一。

攻读博士期间，我非常幸运能与许多优秀的合作者共事，并从他们身上学到了很多：Yegor Klochkov、Naoki Kiyohara、Harrison Zhu、Wenlin Chen、Lapo Rastrelli、Shavindra Jayasekera、Jacob Si、Filippo Valdettaro、Yohan Jung、Hyungi Lee、Thomas M¨ollenhoff、Mohammad Emtiyaz Khan、Bolian Li 和 Ruqi Zhang。我尤其感谢 Yegor Klochkov 和 Mohammad Emtiyaz Khan，为我提供了在字节跳动人工智能实验室和 RIKEN AIP 实习的宝贵机会。Yegor 带我走进了机器学习公平性与优化的世界，而 Emti 则鼓励我跳出固有思维。

感谢攻读博士期间与我有过有趣交谈的每一个人：Carles Balsells Rodas、Zijing Ou、Xavier Sumba Toral、Jiaming Zhang、Wen Wu、Rui Xia、Jean-Francois Ton、Ruocheng Guo、Muhammad Faaiz Taufiq、Zonghao Chen、Yang Liu、Junyu Xuan、Xue Yan、Theodore Papamarkou、Andre Freitas、Danilo S. Carvalho、Hugo Monz´on Maldonado、Marco Miani、Luke Ong，以及许多其他人。我还想向 Yingxue Yang 致以衷心的感谢。我们相识相交二十余年，感谢她在我的闲暇生活中与我分享的所有笑话和难忘时光。

非常感谢 Felipe Tobar 和 Arno Solin 审阅我的学位论文。衷心感谢他们花费时间和精力阅读我的学位论文并提出反馈意见。

Fin此外，我深深感谢我的家人和挚爱之人。我想表达我最深切的 st

g感谢我的父母在这段旅程中始终给予我无条件的支持 y.

摘要

深度序列模型已在广泛的数据模态中取得了卓越成就。尽管其预测性能出色，但部署时的主要顾虑在于缺乏对不确定性的认知。与之相反，概率模型依据概率规则量化未观测变量所关联的不确定性。值得注意的是，贝叶斯方法利用贝叶斯定理，以严谨的方式表达在给定某些观测变量时我们对未观测变量的信念。由于精确贝叶斯推断在大规模情形下计算上不可行，实践中需要采用近似推断。贝叶斯方法的两大瓶颈，尤其是在应用于深度神经网络时，是先验设定和近似质量。在第 3 和 4 章中，我们研究如何利用深度序列模型自身的架构，为概率模型中的先验设定或近似方法选择提供信息。首先，我们基于注意力机制与稀疏高斯过程之间的相似性，开发了一种针对 Transformer 架构量身定制的近似贝叶斯推断方法。接着，我们利用 HiPPOs（高阶多项式投影算子）的长程记忆保留能力，为高斯过程构造跨域诱导点，使其能够在在线学习或持续学习中成功记忆历史序列。除了深度序列模型在预测任务中的进展之外，由一系列潜变量构成的序列生成模型（例如扩散模型）也已在深度生成模型领域广泛普及。受扩散模型中针对这些潜变量的显式自监督信号启发，我们在第 5 章中探讨利用针对其潜在状态的自监督信号改进其他深度生成模型的可能性，并研究序列生成中潜在状态序列所需的概率结构。总体而言，本学位论文利用深度序列模型中的归纳偏置来设计概率推断方法或概率结构，从而弥合深度序列模型与概率模型之间的差距，促使二者相互促进、共同改进。

目录

1 引言 16
1.1 论文概要与贡献 .181.2 发表论文列表 .20

2 背景 22
2.1 深度序列模型架构 .222.1.1 Transformer .22

2.1.2 HiPPO：采用最优多项式投影的循环记忆 252.2 可扩展概率学习 .282.2.1 贝叶斯推断 .282.2.2 变分推断 .292.3 高斯过程 .312.3.1 回归的精确推断 .322.3.2 稀疏变分高斯过程 .342.3.3 深度核学习与深度高斯过程 . . . . . 372.4 深度序列生成模型 .392.4.1 深度潜变量模型 .392.4.2 层次变分自编码器 .412.4.3 扩散模型 .43

3 通过稀疏高斯过程校准Transformer 4 **5**
3.1 引言.463.2 稀疏高斯过程注意力 .473.2.1 作为稀疏变分高斯过程均值的注意力 473.2.2 标准 SGPA 及其在自注意力中的低效性 . . . . . 483.3 通过解耦 SGPA 提高时间与内存效率 . . . . . 493.3.1 解耦稀疏变分高斯过程的预备知识 .503.3.2 解耦 SGPA .513.4 基于解耦 SGPA 的 Transformer .523.5 实验.54

3.5.1 分布内校准 .553.5.2 对分布外数据的稳健预测 . . . . . 573.5.3 分布外检测 .593.5.4 附加结果总结 .613.6 相关工作 .613.7 结论与未来工作 . 62

4 用于在线域间高斯过程的 RecurrentMemory63

4.1 引言.634.2 预备知识 .65

4.2.1 域间高斯过程 .654.2.2 在线稀疏高斯过程 .664.2.3 高斯过程变分自编码器 . . . . . 674.3 基于 HiPPO 的域间诱导点高斯过程 . . . . . 684.3.1 作为域间诱导变量的 HiPPO .684.3.2 随时间调整核矩阵 .684.4 将 OHSVGP 扩展至多维输入 .714.5 相关工作 .744.6 实验.754.6.1 在线时间序列预测 .774.6.2 在 UCI 数据集上进行持续学习 .794.6.3 面向高维时间序列预测的持续学习 854.7 结论. 885

你的图像其实是伪视频的最后一帧 8

95.2 扩散模型与分层变分自编码器.925.2.2 通过伪视频生成改进图像生成 . . 94

5.3 使用伪视频改进 VQ-VAE 的重建与生成

视频 .945.3.1 VQVAE 概述 .955.3.2 实验.965.4 通过高阶马尔可夫伪视频改进生成 . . . . 1005.4.1 一阶马尔可夫链是创建伪视频的最佳选择吗？ . 1005.4.2 实验.1025.5 相关工作 .1055.5.1 序列生成模型 .1055.5.2 自监督学习 .106

5.6 结论与讨论 .10 6

6 结论与未来工作 **108**

6.1 学位论文总结 .1086.2 局限性与未来研究方向 .110

A 第 2 章补充材料 131A.1 除 HiPPO-LegS 之外的 HiPPO 变体 .131

B 第3章补充材料 133

B.1 推导 .133B.1.1 稀疏变分高斯过程（SVGP）的证据下界推导.133B.1.2 基于 SGPA 的 Transformer 证据下界推导 134

B.2 不确定性校准指标 .13 6

B.2.1 适当评分规则 .136B.2.2 校准误差 .136

B.3 补充实验 . 137

B.3.1 标准 SGPA 与解耦方法的实证比较

SGPA 137

B.3.2 使用 ZINC 数据集进行图属性回归 138

B.4 详细实验设置和超参数.139

B.5 运行时间 141

B.6 与稀疏中的稀疏高斯过程的联系 .142

B.7 表格中的结果 143

C 第 4 章的补充材料 15 **1**

C.1 通过直接 ODE 演化计算先验诱导协方差$\mathbf{K}_{\mathbf{uu}}^{(t)}$ 151C.2 后验 OHSVGP 的有限基近似.153C.3 SVGPVAE 模型细节.154C.4 其他结果.155C.4.1 使用可训练核超参数进行时间序列回归的结果.155C.4.2 基函数—测度变体的比较.155

D 第5章的补充材料 15 **7**

D.1 定理 5.4.1 的证明 . 157

D.2 超参数 . 158

图目录

2.1 Transformer 一层中多头自注意力的一个头（h）示意图.24 2.2 HiPPO-LegS 对玩具时间序列（c） $y(x)$进行近似的示意图。这里， x 用于表示任意时间索引。(a) 以结束时间索引 $t = T/2$ 和 $t = T$（此处为$T = 1$ ）为条件的时间相关基函数。(b) 记忆状态 $\boldsymbol{c}(t)$的演化。(d) 以结束时间索引 $t = T/2$和 T为条件的有限基近似$\hat{}y^t(x)$。 . . . . . 25 2.3 稀疏变分高斯过程（SVGP）预测的图模型.35 2.4 使用稀疏高斯过程近似的 L层深度高斯过程预测的图模型.39 2.5 深度潜变量模型的图模型.40 2.6 深度序列生成变量模型的图模型。 . . . . . 41 3.1 下列模型中多头自注意力的一个头（h）示意图：(a) 原始 Transformer、(b) 基于标准 SGPA 的 Transformer，以及 (c) 基于解耦 SGPA 的 Transformer.48 3.2 在 CIFAR10（第 1 行）、CIFAR100（第 2 行）、IMDB（第 3 行）和 CoLA（第 4 行）上训练的 Transformer 或 ViT 的测试集准确率（或 CoLA 的马修斯相关系数）和校准指标.56 3.3 使用数据增强在 CIFAR10（上排）和 CIFAR100（下排）上训练的 ViT 的测试集准确率和校准指标。 . . . . 56 3.4 在 CoLA 上训练的 Transformer 的分布外测试集马修斯相关系数和校准指标.57 3.5 未使用数据增强、在相应干净数据上训练的 ViT，在 CIFAR10-C（上排）和 CIFAR100-C（下排）上的测试集准确率和校准指标随损坏偏斜强度的变化。 57 3.6 使用数据增强、在相应干净数据上训练的 ViT，在 CIFAR10-C（上排）和 CIFAR100-C（下排）上的测试集准确率和校准指标随损坏偏斜强度的变化。 . . . 58

3.7 使用未经数据增强的 CIFAR10 训练的 ViT 进行分布外检测的 AUROC（上）和 AUPR（下）指标。. . . . . 593.8 使用未经数据增强的 CIFAR100 训练的 ViT 进行分布外检测的 AUROC（上）和 AUPR（下）指标。. . . . . 593.9 使用经过数据增强的 CIFAR10 训练的 ViT 进行分布外检测的 AUROC（上）和 AUPR（下）指标。.603.10 使用经过数据增强的 CIFAR100 训练的 ViT 进行分布外检测的 AUROC（上）和 AUPR（下）指标。.604.1 在一个包含 2 个任务的玩具时间序列上应用在线 HiPPO 稀疏变分高斯过程（OHSVGP）。其中 x 用于表示任意时间索引。(a)终止时间索引为$x = t_1$ 和 $x = t_2$的时变基函数。(b) 诱导变量最优近似后验（均值 ± 2 边际标准差）的演变。(c)、(d)、(e)分别展示了后验在线高斯过程、OHSVGP 以及后验 OHSVGP 的有限基重构的预测均值 ± 2 和标准差。.71

4.2 在双月数据集上，采用不同排序准则的 OSVGP 和 OHSVGP 模型在每个任务（共 3 个）之后的决策边界。对于 OSVGP，我们用红色可视化诱导位置。. . . . 724.3 太阳辐照度和音频信号预测数据集上，已学习任务的测试集负对数预测密度与已学习任务数量的关系。 . . . . . 774.4 太阳辐照度和音频信号预测数据集上，已学习任务的测试集均方根误差与已学习任务数量的关系。. . . . . 774.5 预测均值 ± 2OSGPR 、 OVC 、 OVFF 和 OHSGPR 在太阳能数据集任务 10 之后的标准差。M $= 50$ 个诱导变量被使用。 .774.6 在 COVID 数据集的全部 5 个任务上持续学习每个任务后，各任务的测试集负对数预测密度。 .784.7 在 COVID 数据集的全部 5 个任务上持续学习每个任务后，各任务的测试集期望校准误差。 .794.8 在 UCI Skillcraft 数据集的全部 10 个任务上持续学习每个任务后，各任务的测试集负对数预测密度。.814.9 在 UCI Skillcraft 数据集的全部 10 个任务上持续学习每个任务后，各任务的测试集均方根误差。 .824.10 在 UCI 发电厂数据集的全部 10 个任务上持续学习每个任务后，各任务的测试集负对数预测密度。 .834.11 在 UCI 发电厂数据集的全部 10 个任务上持续学习每个任务后，各任务的测试集均方根误差。 .84

4.12 在 ERA5 数据集的全部 10 个任务上持续学习每个任务后，各任务的测试集负对数预测密度。 .864.13 在 ERA5 数据集的全部 10 个任务上持续学习每个任务后，各任务的测试集均方根误差。 .875.1 扩散模型与 HVAE：与标准分层变分自编码器相比，扩散模型利用自监督信号为其中间潜在状态引入归纳偏置。 .925.2 编码器按热方程固定的 HVAE 与采用可学习编码器的标准 HVAE 所生成的数字。两种分层变分自编码器均使用与 Rissanen 等人（2023）相同的解码器架构 . . . . 935.3 通过对狗的图像进行模糊处理而构建的伪视频示例。 .975.4 在 CIFAR10 上训练的 VQVAE/CViViT 的真实图像与重建图像。 .985.5 在 CelebA 上训练的 VQVAE/CViViT 的真实图像与重建图像。 .985.6 使用一阶马尔可夫链（上）和高阶马尔可夫链（下）向 CIFAR10 图像添加高斯噪声所构建的伪视频示例。 .1035.7 分别在 CIFAR10 和 CelebA 的 4帧高阶马尔可夫伪视频上训练的视频扩散模型所生成的图像。104B.1 在 ZINC 上训练的 Transformer 的测试集回归误差和负对数似然指标。 .138B.2 使用在 ZINC 上训练的 Transformer 进行分布外检测的 AUROC 和 AUPR 指标。 .138C.1 太阳辐照度和音频信号预测数据集上，已学习任务的测试集均方根误差与已学习任务数量的关系（持续更新核超参数）。 .155C.2 太阳辐照度和音频信号预测数据集上，已学习任务的测试集负对数预测密度与已学习任务数量的关系（持续更新核超参数）。 .155C.3 在一个玩具在线回归数据集上，基于不同 HiPPO 变体的 OHSGPR 比较。 .156

表格目录

3.1 标准 SGPA 与解耦 SGPA 的复杂度比较 . . . . . 523.2 不同方法在 6 个分布外检测任务中的 AUROC 和 AUPR 平均排名 .58

4.1 时间序列预测实验中，不同模型在单个 NVIDIA RTX3090 GPU 上学习所有任务的墙钟累计运行时间（秒） .79

5.1 采用根据热方程固定的编码器的 HVAE 与采用可学习编码器的标准 HVAE 生成数字的 Inception分数（IS） .9

35.2 C-ViViT（重建）、VideoGPT（自回归生成）和 Phenaki（潜在掩码生成）在由 CIFAR10 与 CelebA 图像构建的伪视频上训练后所生成图像的最后一帧 FID/PSNR。1帧结果来自其图像对应模型 VQVAE（重建）、ImageGPT（自回归生成）和 MaskGit（潜在掩码生成），这些模型在原始 CIFAR10 与 CelebA 图像上训练。.995.3 在由 CIFAR10 与 CelebA 图像构建的伪视频上训练的视频扩散模型所生成图像的最后一帧FID（同时采用一阶马尔可夫或高阶马尔可夫高斯噪声数据增强）。1帧结果来自在原始 CIFAR10 与 CelebA 图像上训练、并采用等效 UNet架构的图像扩散模型。 .1035.4 在由 CIFAR10 与 CelebA 图像构建、并采用高阶马尔可夫高斯噪声数据增强的伪视频上训练的视频扩散模型所生成图像的最后一帧FID。1帧结果来自在原始 CIFAR10 与 CelebA 图像上训练的图像扩散模型。此处，1帧模型使用 4,000 个扩散步数，而 4帧模型总共使用 3,000 个扩散步数。 .105

5.5 在由 CIFAR10 图像构建、并采用高阶马尔可夫数据增强（高斯噪声或高斯模糊）的伪视频上训练的视频扩散模型所生成图像的最后一帧FID。 105

B.1 在 CIFAR10 上训练且未使用数据增强时，基于标准 SGPA 和两种解耦 SGPA 变体的 ViT 测试集准确率和负对数似然。137 B.2 CoLA（批次大小 $= 227$）和 CIFAR10（批次大小 $= 200$）在推断阶段处理一个批次并使用 10 个蒙特卡洛样本的计算时间（单位： s）（结果由单张 Nvidia GTX 2080 Ti 图形处理器卡测得）.141 B.3 SGPA 和最大似然估计在 CoLA（批次大小 $= 32$）和 CIFAR10（批次大小 $= 100$）上每个训练轮次的训练时间（单位： s）（结果由单张 Nvidia GTX 2080 Ti 图形处理器卡测得）.142 B.4 分布内性能：基于 IMDB 的情感分析 . . . . . 143 B.5 分布内性能：基于 CoLA 的语言可接受性 . . . . 143 B.6 分布内性能：未使用数据增强的 CIFAR10 图像分类.144 B.7 分布内性能：使用数据增强的 CIFAR10 图像分类.144 B.8 分布内性能：未使用数据增强的 CIFAR100 图像分类.144 B.9 分布内性能：使用数据增强的 CIFAR100 图像分类 .145 B.10 分布内性能：基于 ZINC 数据集的图属性回归 . 145 B.11 分布外鲁棒性：基于 CoLA 数据集的语言可接受性 . . . . . 145 B.12 未使用数据增强、在干净数据上训练的 ViT 在 CIFAR10-C 上的准确率.145 B.13 使用数据增强、在干净数据上训练的 ViT 在 CIFAR10-C 上的准确率. 146 B.14 未使用数据增强、在干净数据上训练的 ViT 在 CIFAR100-C 上的准确率.146 B.15 使用数据增强、在干净数据上训练的 ViT 在 CIFAR100-C 上的准确率.146 B.16 未使用数据增强、在干净数据上训练的 ViT 在 CIFAR10-C 上的负对数似然.146 B.17 使用数据增强、在干净数据上训练的 ViT 在 CIFAR10-C 上的负对数似然.147

B.18 未使用数据增强、在干净数据上训练的 ViT 在 CIFAR100-C 上的负对数似然。.147B.19 使用数据增强、在干净数据上训练的 ViT 在 CIFAR100-C 上的负对数似然。.147B.20 未使用数据增强、在干净数据上训练的 ViT 在 CIFAR10-C 上的期望校准误差。.147B.21 使用数据增强、在干净数据上训练的 ViT 在 CIFAR10-C 上的期望校准误差。.148B.22 未使用数据增强、在干净数据上训练的 ViT 在 CIFAR100-C 上的期望校准误差。.148B.23 使用数据增强、在干净数据上训练的 ViT 在 CIFAR100-C 上的期望校准误差。.148B.24 未使用数据增强、在干净数据上训练的 ViT 在 CIFAR10-C 上的最大校准误差。.148B.25 使用数据增强、在干净数据上训练的 ViT 在 CIFAR10-C 上的最大校准误差。.149B.26 未使用数据增强、在干净数据上训练的 ViT 在 CIFAR100-C 上的最大校准误差。.149B.27 使用数据增强、在干净数据上训练的 ViT 在 CIFAR100-C 上的最大校准误差。.149B.28 在 CIFAR10 上训练且未使用数据增强的 ViT 用于分布外检测的 AUROC 和 AUPR 指标。.149B.29 在 CIFAR10 上训练且使用数据增强的 ViT 用于分布外检测的 AUROC 和 AUPR 指标。.150B.30 在 CIFAR100 上训练且未使用数据增强的 ViT 用于分布外检测的 AUROC 和 AUPR 指标。.150B.31 在 CIFAR100 上训练且使用数据增强的 ViT 用于分布外检测的 AUROC 和 AUPR 指标。.150B.32 在 ZINC 上训练的 Transformer 用于分布外检测的 AUROC 和 AUPR 指标。.150D.1 C-ViViT 架构和优化器所用的超参数。. . . . 159D.2 VideoGPT 架构和优化器所用的超参数。. . . 159

D.3 Phenaki架构和优化器所用的超参数。. . . . . 159D.4 视频扩散的UNet架构和优化器所用的超参数。.160

第1章

引言

深度学习技术可以追溯到上个世纪（McCulloch 和 Pitts，1943；Rosenblatt，1958），但其复兴始于 2010 年代初期，当时一系列突破相继出现（Krizhevsky 等人，2012；Graves 等人，2013b；Brown 等，2020），证明了深度学习在建模复杂函数以及扩展至大型数据集方面的有效性（LeCun 和 Hinton，2015）。鉴于许多数据模态的输入特征都以序列形式存在（例如自然语言、音频和视频），或可表示为序列（例如，图像可以视为由图像块组成的序列（Dosovitskiy 等，2021）），深度学习近年来的进展在很大程度上由深度序列模型的进步所推动（Vaswani 等，2017；Gu 等人，2020、2022；Gu 和 Dao，2023）。这类模型正逐渐成为适用于不同模态的统一建模范式，并不断突破我们对其能力上限的预期（Jumper 等人，2021；OpenAI 等人，2024b；Zeni 等人，2024）。

深度序列模型的关键组成部分之一，是若干为序列建模精心设计的网络架构。基于注意力架构的 Transformer（Vaswani 等，2017）已在诸多领域取得成功，包括计算机视觉（Dosovitskiy 等，2021）、自然语言处理（Devlin 等人，2019；Brown 等，2020；OpenAI 等人，2024b）、语音识别（OpenAI 等人，2024a），以及在科学领域中的应用（Jumper 等人，2021；Zeni 等人，2024）。此外，深度状态空间模型（状态空间模型；（Gu 等人，2020、2022；Gu 和 Dao，2023））最初旨在高效建模具有广泛上下文的序列中的长期依赖，为 Transformer 提供了有竞争力的替代方案，并已被部署于越来越多的应用中（Zhu 等人，2024；Li 等人，2024；Quan 和 Li，2024）。

在深度生成建模领域，序列生成模型（Sønderby 等人，2016年；Ho 等人，2020年）通过引入一系列潜变量扩展了单潜变量模型，从而提升了模型的表达能力。这些模型不再一步将噪声生成为目标域中的数据，而是通过多个步骤逐渐将噪声转换为目标数据。

其中，扩散模型（Sohl-Dickstein 等人，2015年；Ho 等人，2020年；Song 等人，2021b）及其变体（Kingma 等人，2021年；Nichol 和 Dhariwal，2021年；Song 等人，2021a；Rissanen 等人，2023年；Bansal 等人，2023年；Hoogeboom 和 Salimans，2023年）在生成照片级真实感图像方面表现出色；除视觉之外，这些模型也已应用于其他数据模态（Benita 等人，2023年；Campbell 等人，2023年；Lou 等人，2024年）。

尽管深度序列模型已成为现代机器学习系统的基础构件，但我们仍需认识到，它们并非所有应用场景的终极解决方案。例如，这些模型往往不了解自身知识的局限，在预测中给出的不确定性也不可靠，因此无法部署于安全关键型应用（Xiong 等人，2024年；Wen 等人，2024年）。

另一方面，基于概率定律和贝叶斯统计的概率方法（Bayes，1763）提供了一个严谨的框架，用于推理不确定性并研究随机变量的结构。在这一框架中，我们从代表对未知事物初始信念的先验分布出发，并利用观测数据中的信息将先验更新为后验分布，通常借此降低不确定性，这一过程通过贝叶斯定理实现。此外，该框架非常适用于序贯学习或不确定性下的决策。每当有新数据到来时，都可以将旧的后验分布视为新的先验，从而执行贝叶斯更新。

将概率方法的互补优势巧妙地融入深度序列模型，或许能够解决与之相关的一些根本性问题，包括不可靠的不确定性估计。过去十年间，人们开展了许多工作，试图利用概率方法提升深度学习的可靠性，由此形成了一个重要的研究领域，即贝叶斯深度学习（Wilsonand Izmailov，2022年；Arbel 等人，2023年；Papamarkou et al.，2024年；Chen 等人，2025c）。然而，将贝叶斯统计应用于深度学习时，其两个主要瓶颈会被进一步放大。首先，有意义的先验和模型设定颇具挑战，尤其是对于贝叶斯神经网络而言。由于深度神经网络具有高度非线性且参数过度化，为其权重指定的先验对所诱导函数的影响并不直观（Sunet al.，2019年；Ma and Hern´andez-Lobato，2021年）。其次，在实践中，近似贝叶斯推断（Jordan etal.，1999；Brooks etal.，2011年；Li，2018年）不可或缺，因为精确后验需要

难解积分。这引发了关于近似后验的选择与结构的问题，而理想的近似需要同时具备准确性和可扩展性。

本学位论文旨在推动贝叶斯深度学习在深度序列模型领域的发展。如今，深度序列模型已成为广泛应用中的统一建模工具。我们不再仅从概率视角思考解决方案，例如手工为无数网络权重构造有意义的先验，而是通过利用深度序列模型中的归纳偏置（如网络架构），在其取得的巨大进展基础上继续探索。我们利用这些归纳偏置，构建适用于这类模型的概率模型和近似技术。这不仅能通过结合深度学习与概率建模进一步提升效果，还能确保本文提出的方法保留深度序列模型的特征机制。

1.1 学位论文概要与贡献

本学位论文其余章节安排如下。

- 第2章介绍本学位论文的技术背景。我们回顾深度序列模型、概率建模和贝叶斯深度学习中的关键概念与思想，这些构成了本文所提方法的基础。本章最后介绍深度概率建模的两个核心应用，即不确定性量化和深度生成模型。

- 第3 章旨在通过揭示点积注意力与稀疏高斯过程（SGP；（Snelson 和 Ghahramani，2005））后验均值之间的联系，校准 Transformer 模型（Vaswani 等人，2017年）的不确定性。基于这一关键联系，我们提出稀疏高斯过程注意力（SGPA），以有效的对称核替代 Transformer 中的缩放点积操作，并直接运用稀疏高斯过程技术来近似多头注意力块（MHAs）输出空间中的后验过程。在文本、图像和图结构的一系列预测任务上的实证结果表明，基于 SGPA 的 Transformer 在实现具有竞争力的预测准确率的同时，显著改善了分布内不确定性校准、分布外鲁棒性及检测能力。本章基于Chen 和 Li (2023年)

- 第4章提出了一种新颖的在线高斯过程（GP）模型（Bui等人，2017年），用于在在线学习设置下捕捉对序列观测数据的长期记忆。我们的模型——在线 HiPPO 稀疏变分高斯过程

（OHSVGP）利用 HiPPO框架（Gu 等人，2020年）的长程记忆建模能力，将 HiPPO 时变正交投影视为能够记忆过程历史序列的诱导变量。我们证明，HiPPO框架自然适用于跨域高斯过程框架（Van der Wilk 等人，2020年），并展示了核矩阵也可以根据 HiPPO 的 ODE演化以递推形式在线更新。我们通过时间序列的一维时间序列在线预测、对具有多维输入的数据的判别式高斯过程模型进行持续学习，以及使用稀疏高斯过程变分自编码器进行深度生成建模（Jazbec 等人，2021年）来评估 OHSVGP，结果表明，与现有在线高斯过程方法相比，该模型在预测性能、长期记忆保持和计算效率方面表现更佳。本章基于Chen 等人（2025b）

- 第 5 章的动机源于扩散模型（Sohl- Dickstein 等人，2015年；Ho 等人，2020年；Song 等人，2021b）相较于其他序列生成模型（例如层次变分自编码器（HVAEs；（Sønderby 等人，2016年；Maaløe 等人，2019年；Vahdat 和 Kautz，2020年））的优异表现。我们推测，这在一定程度上可归因于受损图像为模型的中间潜在状态提供了额外的自监督信息；受损图像与原始图像共同构成伪视频。基于这一假设，我们探究利用此类伪视频改进其他类型生成模型的可能性：首先将图像生成模型扩展为相应的视频生成模型，然后对原始图像应用数据增强，构造伪视频，并以此训练视频生成模型。此外，我们从概率视角分析扩散模型中常用的一阶马尔可夫数据增强方法在序列生成中可能存在的问题，并提出采用具有更强表达能力的概率结构的数据增强方法，以便在伪视频中构造更有用的信息。我们通过在 CIFAR10 和 CelebA 数据集上进行实验，实证验证了伪视频所提供的额外自监督信息的有效性。本章内容基于 Chen 等人（2025a）

- 第6 章展望了本学位论文的主要发现与贡献。此外，我们还指出了若干开放性挑战以及未来工作的可行方向。

1.2 出版物列表

本节完整列出了我在攻读博士期间参与合著的论文。对于内容收录于本学位论文中的论文，标题以粗体标示；我还简要介绍了自己对这些论文的贡献。本学位论文中的部分材料，包括观点、概念、文本、图和表，已在这些论文中发表。上标星号（*）表示共同第一作者且贡献相同。

同行评审会议论文。

1. (Chen 等人, 2025b) Wenlong Chen*、Naoki Kiyohara*、Harrison Bo Hua Zhu*、Jacob Curran-Sebastian、Samir Bhatt和Yingzhen Li。在线跨域高斯过程的循环记忆。载于《神经处理系统进展》（NeurIPS），2025。我的贡献：主要想法由我提出，末位作者提供了有益建议。三位共同第一作者（*）均在末位作者的指导下参与了代码实现、实验和论文撰写。此外，我完成了全部推导，并协助统筹其他作者的贡献。2. (Jayasekera 等人, 2025) I. Shavindra Jayasekera*、Jacob Si*、Filippo Valdettaro、Wenlong Chen、A. Aldo Faisal和Yingzhen Li。上下文学习的变分不确定性分解。载于《神经处理系统进展》（NeurIPS），2025。3. (Jung 等人, 2025a) Yohan Jung、Hyungi Lee、Wenlong Chen、Thomas M¨ollenhoff、Yingzhen Li、Juho Lee、Mohammad Emtiyaz Khan。持续逻辑回归的紧凑记忆。载于《神经处理系统进展》（NeurIPS），2025。4. (Chen 等人, 2024年) Wenlong Chen*、Yegor Klochkov*和Yang Liu。事后偏差评分对公平分类是最优的。载于《国际学习表征会议》（ICLR），2024年。获评Spotlight报告。5. (Chen and Li, 2023年) Wenlong Chen和Yingzhen Li。通过稀疏高斯过程校准Transformer。载于《国际学习表征会议》（ICLR），2023年。我的贡献：主要想法由我提出，末位作者提供了有益建议。手稿主要由我撰写，末位作者协助润色。此外，我编写了全部代码，并完成了这项工作的所有实验。

经同行评审的研讨会论文。

1.（Chen 等人，2025a）Wenlong Chen*，Wenlin Chen*，Lapo Rastrelli 与 Yingzhen Li。**你的图像其实是伪视频的最后一帧**。收录于机器学习中的深度生成模型：理论、原理与效能（*DeLTa*）*ICLR* 研讨会，2025。我的贡献：主要想法由我提出，末位作者提供了有益建议。两位共同第一作者（∗）在末位作者的指导下参与了代码实现、实验和论文撰写。第三作者协助完成了与视频扩散模型相关的实验。我还协助统筹了其他作者的贡献。2.（Jung 等人，2025b）Yohan Jung*，Hyungi Lee*，Wenlong Chen*，Thomas M¨ ollenhoff、Yingzhen Li、Juho Lee 与 Mohammad Emtiyaz Khan。用于基于 K-prior 的持续学习的紧凑记忆。收录于近似贝叶斯推断进展研讨会（*AABI*），2025。

预印本。

1. (Chen 等人，2025c) Wenlong Chen*，Bolian Li*，Ruqi Zhang 和 Yingzhen Li。《深度学习中的贝叶斯计算》。载于*arXiv* 预印本 *arXiv:2502.18300*，2025 年（即将作为章节收录于《马尔可夫链蒙特卡罗手册——第二版》）。

第 2 章

背景

作为两个快速发展的研究领域，深度序列模型和概率建模都积累了大量文献。本章旨在简要介绍这两个领域中的相关概念和方法，并以此为基础开展本学位论文的研究。我们首先回顾深度序列建模中的两种流行网络架构，即 Transformer（第2.1.1节），以及为深度状态空间模型（SSM）奠定理论基础的高阶多项式投影算子（HiPPO）（第2.1.2节）。接下来，我们介绍概率机器学习中的关键概念，以及用于近似后验分布的变分推断（第2.2.2节）。我们将详细回顾一类经典的概率模型——高斯过程，以及利用它们构建深度模型的常用技术（第2.3节）。除了预测模型，我们最后还将回顾概率机器学习在深度生成模型领域中的应用，尤其关注序列生成模型（第2.4节）。

2.1 深度序列模型架构

如今，序列建模是许多机器学习应用的核心，而深度学习在这一领域取得的巨大成功，很大程度上归功于几种强大的网络架构。本节将回顾两种流行的深度序列模型架构：Transformer 和高阶多项式投影算子（HiPPO）。

2.1.1 Transformer

通用Transformer由多层感知机（MLP）和多头自注意力（MHSA）构成，下面我们回顾这两种基础架构。

多层感知机。 多层感知机（MLP；（Rosenblatt，1958））是深度神经网络中无处不在的基础组件。给定输入 $\boldsymbol{x} \in \mathbb{R}^{d_{in}}$，一个 L层多层感知机 $f_{\boldsymbol{\theta}}(\boldsymbol{x})$：$\mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ 对输入进行如下一系列变换：

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{W}^L g(\mathbf{W}^{L-1} g(\cdots g(\mathbf{W}^1 \boldsymbol{x} + \boldsymbol{b}^1)) + \boldsymbol{b}^{L-1}) + \boldsymbol{b}^L, \tag{2.1}$$

其中，我们用 $\boldsymbol{\theta} = \{\mathbf{W}^l,\ \boldsymbol{b}^l\}_{l=1}^L$ 表示可学习权重矩阵和偏置向量的集合，

$$\mathbf{W}^1 \in \boldsymbol{R}^{d_h \times d_{in}}, \boldsymbol{b}^1 \in \boldsymbol{R}^{d_{in}}, \mathbf{W}^l \in \boldsymbol{R}^{d_h \times d_h}, \boldsymbol{b}^l \in \boldsymbol{R}^{d_h}, l = 2, ..., L-1,$$
$$\mathbf{W}^L \in \boldsymbol{R}^{d_{out} \times d_h}, \boldsymbol{b}^L \in \boldsymbol{R}^{d_{out}}.$$

这里，该多层感知机的隐藏维度（或宽度）和输出维度分别为d_h 和 d_{out}。 $g(\cdot)$ 是逐元素非线性激活函数，使多层感知机关于 $\boldsymbol{x}$ 成为非线性函数。其中，修正线性单元（ReLU；（Nair 和 Hinton，2010））及其变体（Klambauer 等人，2017年）是最常用的激活函数。ReLU 是一种分段函数，将正输入映射为其自身，将负输入映射为 0：

$$\mathrm{ReLU}(s) := \max(0, s). \tag{2.2}$$

尽管大多数现代神经网络架构都比纯多层感知机更复杂，但它们仍常被用作现代架构中的基本模块，例如用作表达能力强的神经特征提取器。

多头注意力。 注意力机制最早由Graves 等人(2013a) 提出，现已成为 Transformer 模型的标志性构建模块。现代 Transformer 中最广泛使用的注意力模块是基于点积的注意力（Vaswani 等人，2017年；Dosovitskiy 等人，2021年）。给定一个由 T_q查询 $\boldsymbol{q} \in \mathbb{R}^{T_q \times d_q}$组成的序列、一个由 T_k （通常 $T_q = T_k := T$ 如下面讨论的自注意力中的情形）键 $\boldsymbol{k} \in \mathbb{R}^{T_k \times d_k}$ ($d_k = d_q$组成的序列，以及与这些键相关联的值 $\boldsymbol{v} \in \mathbb{R}^{T_k \times d_v}$，点积注意力（Vaswani 等人，2017年）的计算方式如下：

$$\boldsymbol{F} = \omega(\boldsymbol{q}\boldsymbol{k}^\top)\boldsymbol{v}, \tag{2.3}$$

其中，ω 是一个非线性激活函数。点积注意力计算值的加权和，其中权重可视为查询与键之间的相似度评分，由它们的点积衡量，$\boldsymbol{q}\boldsymbol{k}^\top$。对于自注意力，键直接设为与查询相同，即 $\boldsymbol{k} = \boldsymbol{q}$。Transformer 采用

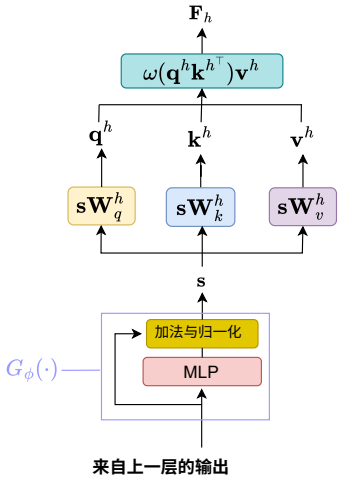


图 2.1：Transformer 一层中多头自注意力的一个头（h）示意图。

多头自注意力（MHSA）由多个点积注意力头构成，其结果经过聚合后生成最终输出。每个注意力头的查询、键和值均由同一输入序列得到。具体而言，MHSA 对点积自注意力作如下改动。假设 H 个注意力头，给定一段长度为 T 的输入 $\boldsymbol{s} \in \mathbb{R}^{T\times d_s}$，输入多头自注意力模块后，我们通过投影矩阵 h 将其投影为各个注意力头的查询 $\boldsymbol{W}_q^h \in \mathbb{R}^{d_s\times d_q}$：$\boldsymbol{q}^h = \boldsymbol{s}\boldsymbol{W}_q^h$。相应地，我们使用矩阵 $\boldsymbol{k}^h$ 和 $\boldsymbol{v}^h$ 进行投影，得到键 $\boldsymbol{W}_k^h \in \mathbb{R}^{d_s\times d_k}$ 和值 $\boldsymbol{W}_v^h \in \mathbb{R}^{d_s\times d_v}$。通常，所有注意力头使用相同的$d_q = d_k = d_v$。然后，将 $\boldsymbol{F}_h$ 代入$\boldsymbol{q}^h$、 $\boldsymbol{k}^h$ 和 $\boldsymbol{v}^h$，即可根据公式 2.3得到该注意力头的输出。最后，使用输出投影矩阵 $\boldsymbol{W}_F \in \mathbb{R}^{(Hd_v)\times(Hd_v)}$，将各个注意力头的注意力输出合并如下：

$$\boldsymbol{F} = \text{concat}(\boldsymbol{F}_1, \cdots, \boldsymbol{F}_H)\boldsymbol{W}_F. \tag{2.4}$$

在 Transformer 中，可以将多个 MHSA 层堆叠在一起，其中第（$l-1$）个 MHSA 层的输出会进一步经过一个非线性函数处理G_{ϕ^l}——该函数由多层感知机参数化，并采用常见的深度学习技巧（例如残差连接（He 等人，2015年）和层归一化（Ba 等人，2016年））——以获得第 l 个 MHSA 层的输入，即 $\boldsymbol{s}^l = G_{\phi^l}(\boldsymbol{F}^{l-1})$。图 2.1 展示了 Transformer 模型中的一个 MHSA 模块（不包括公式 2.4中的组合投影步骤）。

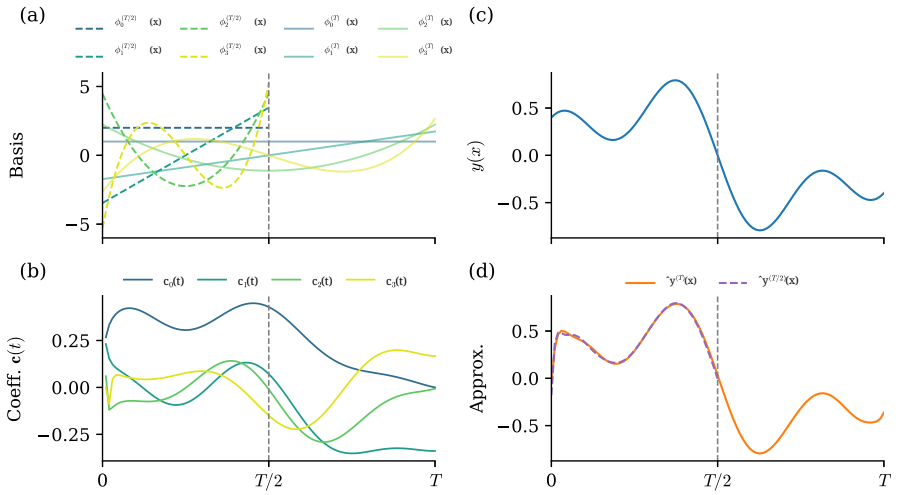


图 2.2：HiPPO-LegS 对玩具时间序列 (c)$y(x)$ 进行近似的示意图。这里用 x 表示任意时间索引。(a) 终止时间索引为 $t = T/2$ 、此处为 $t = T$ ($T = 1$) 时的时变基函数。(b) 记忆状态 $\boldsymbol{c}(t)$ 的演变。(d) 终止时间索引为$t = T/2$、 T时的有限基近似 $\hat{}y^t(x)$。

2.1.2 HiPPO：采用最优多项式投影的循环记忆

HiPPO框架（Gu 等人，2020）旨在构建一种紧凑表示，以捕捉序列中的长期依赖，这是长序列建模的一大挑战。HiPPO为通过最优多项式投影将连续时间信号压缩为有限维记忆状态提供了数学基础。给定一个时间序列 $y(t)$，HiPPO维护一个记忆状态$\boldsymbol{c}(t) \in \mathbb{R}^M$，该状态能够对历史信号 $\{y(x)\}_{x \le t}$进行最优近似。该框架包含一个定义输入重要性的时变测度 $\omega^{(t)}(x)$，其定义域为（$-\infty$， t] ，以及自适应多项式基函数 $\{g_m^{(t)}(x)\}_{m=0}^{M-1}$；这些基函数通过调整基函数的定义域以覆盖新的时间区域而获得。例如，HiPPO-LegS 变体中使用的标准勒让德多项式 $P_m(x)$ 的定义域为 $[-1, 1]$；当最后一个历史时刻为 t （假设其为正数）时，我们定义自适应基

$$g_m^{(t)}(x) := (2m+1)^{1/2} P_m\left(\frac{2x}{t} - 1\right), \tag{2.5}$$

其定义域为 $[0, t]$。此外，这些自适应基在某些 $\omega^{(t)}(x)$ 下是正交归一的（例如，对于上文的自适应勒让德多项式，相应的$\omega^{(t)}(x)$ 在 $[0, t]$ 上均匀分布）：

$$\int_{-\infty}^{t} g_m^{(t)}(x) g_l^{(t)}(x) \omega^{(t)}(x) \mathrm{d}x = \delta_{ml}. \tag{2.6}$$

历史信号通过以下投影系数进行编码

$$c_m(t) = \int_{-\infty}^{t} y(x) g_m^{(t)}(x) \omega^{(t)}(x) \mathrm{d}x, \tag{2.7}$$

从而得到近似

$$y(x) \approx \hat{y}^{(t)}(x) = \sum_{m=0}^{M-1} c_m(t) g_m^{(t)}(x), \tag{2.8}$$

对于 $x \in (-\infty, t]$。除非 $y(x)$ 位于有限基函数的张成空间中，否则压缩通常会造成信息损失。不过，$\{c_m(t)\}_{m=0}^{M-1}$ 在公式2.7中得到的结果是最优的，因为它使以下 L^2-误差最小：

$$\{c_n\}_{n=0}^{M-1} = \underset{\{c'_n\}_{n=0}^{M-1}}{\arg\min} \int_{-\infty}^{t} \|y(x) - \sum_{n=0}^{M-1} c'_n(t) g_n(x)\|^2 \omega^{(t)}(x) \mathrm{d}x. \tag{2.9}$$

因此，M 维记忆状态 $\boldsymbol{c}(t) := [c_0(t) \ldots, c_{M-1}(t)]^\top$ 构成了一种紧凑表示，能够以最优方式捕获 $y(x)$ 中截至 $x = t$ 的历史信息。

该记忆状态 $\boldsymbol{c}(t)$ 是历史序列 t 终点的函数。每当新的时间点出现观测值 $t + \Delta t$ 时，我们便针对 $t + \Delta t$ 调整基函数，并计算 $\boldsymbol{c}(t + \Delta t)$。记忆状态的一个优良特性是支持在线更新：每次观测到新数据时，我们无需从头重新计算积分来获取记忆状态，而是可以根据 $\boldsymbol{c}(t)$ 在线更新得到 $\boldsymbol{c}(t + \Delta t)$。HiPPO 中考虑的自适应多项式基函数 $\frac{\partial}{\partial t} g_m^{(t)}(x)$ 的时间导数可以表示为低阶多项式基的线性组合 $\{g_l^{(t)}(x)\}_{l=0}^{m-1}$。因此，$\boldsymbol{c}(t)$ 关于 t 的时间导数满足一个线性常微分方程（ODE）

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{c}(t) = \boldsymbol{A}(t) \boldsymbol{c}(t) + \boldsymbol{B}(t) y(t), . \tag{2.10}$$

对线性常微分方程进行离散化，可得到如下形式的递推

$$\boldsymbol{c}_k = \boldsymbol{A}_k \boldsymbol{c}_{k-1} + \boldsymbol{B}_k y_k, \tag{2.11}$$

这使得无需重新访问过去的信号即可进行在线更新。这里，$\boldsymbol{A}_k$ 和 $\boldsymbol{B}_k$ 由 $\boldsymbol{A}(t)$、$\boldsymbol{B}(t)$ 以及离散化步长决定。

HiPPO 支持多种测度-基函数配置，这些配置能够对相应的记忆状态进行在线更新（Gu 等人，2020年，2023年）。在此，我们给出一种典范实例化 HiPPO-LegS 的若干细节。对于给定的结束时间索引 t，

HiPPO-LegS 使用均匀测度 $\omega^{(t)}(x) = \frac{1}{t}\mathbf{1}_{[0,t]}(x)$，以及将输入域适配到 $[0,t]$, $g_m^{(t)}(x) = (2m+1)^{1/2}P_m\left(\frac{2x}{t}-1\right)$ 的缩放勒让德多项式作为基函数，其中 $P_m(\cdot)$ 是输入域为 $[-1,1]$ 的标准第 m 个勒让德多项式， $\mathbf{1}_{[0,t]}(x)$ 是区间 $[0,t]$ 上的指示函数。过去上的均匀测度提供了归纳偏置，促使 HiPPO-LegS 将整个历史保留在记忆中。HiPPO-LegS 的记忆状态按照式 2.10 中的线性常微分方程演化。为说明这一点，我们定义 $\phi_m^{(t)}(x) := g_m^{(t)}(x)\omega^{(t)}(x)$。可以证明， $\phi_m^{(t)}(x)$ 的时间导数为：

$$\begin{aligned}\frac{\partial}{\partial t}\phi_m^{(t)}(x) &= \omega_m^{(t)}(x)\frac{\partial}{\partial t}g_m^{(t)}(x) + g_m^{(t)}(x)\frac{\partial}{\partial t}\omega_m^{(t)}(x) \\ &= -\frac{\sqrt{2m+1}}{t}[\frac{m+1}{\sqrt{2m+1}}\phi_m^{(t)}(x) + \sqrt{2m-1}\phi_{m-1}^{(t)}(x) \\ &\qquad + \sqrt{2m-3}\phi_{m-2}^{(t)}(x) + \cdots] + \frac{1}{t}\,\delta_t(x),\end{aligned} \tag{2.12}$$

其中 $\delta_t(x)$ 是位于 $x=t$ 处的狄拉克 δ 函数，它源自均匀测度的时间导数 $\omega^{(t)}(x)$（详情请参见 Gu 等人 (2020年的附录 D.3）。将此表达式代入

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{c}(t) &= [\frac{\mathrm{d}}{\mathrm{d}t}c_0(t), \cdots, \frac{\mathrm{d}}{\mathrm{d}t}c_{M-1}(t)]^\top \\ &= [\int y(x)\frac{\partial}{\partial t}\phi_0^{(t)}(x)dx, \cdots, \int y(x)\frac{\partial}{\partial t}\phi_{M-1}^{(t)}(x)dx]^\top\end{aligned} \tag{2.13}$$

我们可以据此确定记忆状态常微分方程更新（式 $\boldsymbol{A}(t)$ 中的矩阵 $\in \mathbb{R}^{M\times M}$ 和 $\boldsymbol{B}(t) \in \mathbb{R}^{M\times 1}$ 。对于 HiPPO-LegS，它们的显式公式为：2.10

$$\boldsymbol{A}(t) = \frac{1}{t}\boldsymbol{A}, \quad [\boldsymbol{A}]_{mk} = \begin{cases} -\sqrt{(2m+1)(2k+1)} & \text{if } m > k \\ -m-1 & \text{if } m = k \\ 0 & \text{if } m < k \end{cases} \tag{2.14}$$

and

$$[\boldsymbol{B}(t)]_m = \frac{[\boldsymbol{B}]_m}{t} = \frac{\sqrt{2m+1}}{t}, \tag{2.15}$$

其中，因子 $1/t$ 反映了基函数随时间变化而缩放至自适应区间 $[0,t]$。图 2.2 展示了 HiPPO-LegS 中一个玩具时间序列的记忆状态演变，以及根据记忆状态和自适应勒让德多项式对该时间序列进行的近似；其中历史序列采用了两个不同的结束时间索引。

We 介绍使用不同组合的其他 HiPPO 变体的详细信息 f

附录中关于函数基和测度的内容 A.1。与 HiPPO-LegS 相比，它们采用的测度更重视近期历史，因此不太适用于需要在序列中保留长期记忆的任务。

近期的一些研究扩展了 HiPPO递推，以更高效地进行长程记忆建模，并将其与多层感知机堆叠起来，构建深层状态空间模型；这类架构如今已能与 Transformer 相媲美。结构化状态空间序列（S4）模型（Gu 等人，2022年）通过引入可训练的递推参数扩展了 HiPPO，并利用卷积核加速计算；而 Mamba（Gu 和 Dao，2023年；Dao 和 Gu，2024年）则进一步引入了硬件感知选择性状态机制。

2.2 可扩展的概率学习

概率机器学习以概率规则和贝叶斯统计为基础，提供了一个原则严谨的框架，使我们能够在不确定性下系统地表达并更新对未知事物的认知。具体而言，贝叶斯推断利用贝叶斯定理（Bayes，1763），在观测到数据后更新我们对未观测变量的信念。

2.2.1 贝叶斯推断

给定一个由某些参数 $\boldsymbol{\theta}$ 指定的机器学习模型（例如，深度神经网络的权重），我们可以将参数视为未观测的随机变量，并在其上设定先验分布 $p(\boldsymbol{\theta})$，从而将模型转化为概率模型。先验分布表示在观测到任何数据之前，我们对模型参数的初始信念。例如，它可以用于排除可能导致不稳定预测的极端参数值。数据 $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{n=1}^{N}$ 被视为观测变量，而来自数据的关于 $\boldsymbol{\theta}$ 的信息（也称为似然）$p(\mathcal{D}|\boldsymbol{\theta})$，则用于将先验分布更新为后验分布：

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta}, \mathcal{D})}{p(\mathcal{D})} = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{2.16}$$

由后验分布得到的 $\boldsymbol{\theta}$ 模型不确定性，随后通过以下边缘化定义的后验预测分布，转化为与对新输入 $\boldsymbol{x}^*$进行预测相关的预测不确定性：

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}. \tag{2.17}$$

除了具备不确定性量化能力之外，贝叶斯推断还天然适用于在线学习（Cesa-Bianchi 和 Lugosi，2006年；Ritter 等人，2018年）或持续学习（Kirkpatrick 等人，2017年；Nguyen 等人，2018年），在这类学习中，数据点会按批次依次被观测到。当新数据到达时，基于旧数据获得的先前后验分布，$p(\boldsymbol{\theta}|\mathcal{D}_{old})$，会成为新的先验；然后我们结合当前数据 $\mathcal{D}_{new}$，再次通过贝叶斯定理形成新的后验分布：

$$p(\boldsymbol{\theta}|\mathcal{D}_{new},\mathcal{D}_{old}) = \frac{p(\boldsymbol{\theta}|\mathcal{D}_{old})p(\mathcal{D}_{new}|\boldsymbol{\theta},\mathcal{D}_{old})}{\int p(\boldsymbol{\theta}|\mathcal{D}_{old})p(\mathcal{D}_{new}|\boldsymbol{\theta},\mathcal{D}_{old})d\boldsymbol{\theta}}, \tag{2.18}$$

每当有新数据到来时，都可以重复这一过程，从而在不确定性下实现在线学习或持续学习。

遗憾的是，如式2.16 所示，精确贝叶斯推断通常难以计算，因为它需要对分母中的边际似然求解复杂积分，$p(\mathcal{D}) = \int p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})d\boldsymbol{\theta}$；除非先验与似然共轭，否则该积分没有解析解。因此，在实践中，近似推断技术（Li，2018年）不可或缺；这类技术用近似后验 $q(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta}|\mathcal{D})$ 来近似目标后验分布。在这种情况下，式2.17中的后验预测分布也可以基于 $q(\boldsymbol{\theta})$ 近似如下：

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*,\mathcal{D}) \approx \int p(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta})q(\boldsymbol{\theta})d\boldsymbol{\theta} \approx \frac{1}{M}\sum_{m=1}^{M} p(\boldsymbol{y}^*|\boldsymbol{x}^*,\boldsymbol{\theta}_m), \quad \boldsymbol{\theta}_m \sim q(\boldsymbol{\theta}), \tag{2.19}$$

其中，通常会进一步使用蒙特卡洛估计进行边缘化。

在本学位论文中，我们采用变分推断（Jordan 等人，1999；Beal，2003；Li，2018年；Zhang 等人，2018a），这是一种基于优化的近似推断技术，其可扩展性远高于基于模拟的技术，例如马尔可夫链蒙特卡罗（Metropolis 等人，1953；Gelfand，2000；Ma 等人，2015年）。因此，变分推断成为深度学习应用中一种常用的近似推断方法，因为在这类应用中，可扩展性至关重要。

2.2.2 变分推断

变分推断（VI）（Beal，2003；Jordan 等人，1999）将推断问题转化为优化过程。首先指定一个参数分布族，$\mathcal{Q}:=\{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\}$，其参数 $\boldsymbol{\phi}$可调，例如因子化高斯分布，其中 $\boldsymbol{\phi}$是均值和方差参数。然后对 $\mathcal{Q}$ 进行优化，以找出最佳近似后验（也称最优变分分布），即 $q_{\boldsymbol{\phi}^*}(\boldsymbol{\theta})$，它是在参数分布族中与目标后验分布 $p(\boldsymbol{\theta}|\mathcal{D})$ 最接近的分布，其接近程度由 Kullback-Leibler（KL）散度衡量。具体而言，该过程需要针对所选参数 $\boldsymbol{\phi}$ 进行

参数分布族进行优化，以获得最优变分分布 ：

$$\begin{aligned}\boldsymbol{\phi}^* &= \arg\min_{q_{\boldsymbol{\phi}} \in \mathcal{Q}} \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \| p(\boldsymbol{\theta}|\mathcal{D})), \\ \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \| p(\boldsymbol{\theta}|\mathcal{D})) &= \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathcal{D})} d\boldsymbol{\theta}.\end{aligned} \tag{2.20}$$

随后，我们可以将 $q_{\boldsymbol{\phi}^*}(\boldsymbol{\theta})$代入式 $q(\boldsymbol{\theta})$2.19 中，以得到近似后验预测分布。

然而，KL 散度需要计算 $p(\boldsymbol{\theta}|\mathcal{D})$，而它依赖于难解边缘似然（也称为模型证据） $p(\mathcal{D})$，这正是最初采用近似推断的原因。因此，无法直接最小化 KL 散度，需要采用另一种目标函数来进行最小化，由此引出以下作为目标函数的证据下界（ ELBO）。ELBO 是对数边际似然的一个下界，通过从对数边际似然中减去 KL 散度得到；对数边际似然相对于 $\boldsymbol{\phi}$ 是一个常数。因此，可以通过最大化（ELBO）间接实现 KL 散度的最小化。

$$\begin{aligned}\mathcal{L}_{ELBO}(\boldsymbol{\phi}) &:= \log p(\mathcal{D}) - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \| p(\boldsymbol{\theta}|\mathcal{D})) \\ &= \log p(\mathcal{D}) - \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) p(\mathcal{D})}{p(\boldsymbol{\theta}) p(\mathcal{D}|\boldsymbol{\theta})} d\boldsymbol{\theta} \\ &= \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta}) p(\mathcal{D}|\boldsymbol{\theta})}{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})} d\boldsymbol{\theta} \\ &= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[\log p(\mathcal{D}|\boldsymbol{\theta})]}_{\text{ELL}} - \underbrace{\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \| p(\boldsymbol{\theta}))}_{\text{KL regularizer}}\end{aligned} \tag{2.21}$$

上述 ELBO 可分解为两项。第一项通常称为对数似然期望（ELL），第二项则是近似后验 $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ 与先验 $p(\boldsymbol{\theta})$之间的 KL 正则项。ELL 项鼓励$q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$更加重视能够解释数据的模型配置，而第二项则将 $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$正则化，使其趋向先验。两项结合起来，平衡了数据拟合与正则化。

在实践中，除非先验分布、似然和变分分布的设定满足某些特殊条件（例如高斯性），否则我们可能无法对证据下界中的一项或两项进行解析计算。因此，通常还会采用蒙特卡洛估计：

$$\mathcal{L}_{ELBO} = \frac{1}{M} \sum_{m=1}^{M} \left[\log p(\mathcal{D}|\boldsymbol{\theta}_m) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}_m) + \log p(\boldsymbol{\theta}_m)\right], \qquad \boldsymbol{\theta}_m \overset{\text{独立同分布}}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{\theta})。 \tag{2.22}$$

如果所选的变分分布族 $\mathcal{Q}$包含真实的后验分布，那么在优化得到完美求解的情况下，变分推断将返回该后验分布。在实践中，出于计算预算以及分布族可处理性的考虑，通常会采用简单而易于处理的 $\mathcal{Q}$ ，以实现快速计算并降低内存复杂度。

为了利用基于梯度的优化来优化证据下界，重参数化技巧(Kingma 和 Welling, 2014年) 被提出，用于多种变分分布族，包括高斯分布。这种方法将采样操作 $\boldsymbol{\theta} \sim q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ 视为将辅助噪声变量 $\boldsymbol{\phi}$ 通过函数 $\boldsymbol{\epsilon} \sim p_{base}(\boldsymbol{\epsilon})$ 传递，从而使其关于 $T_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})$ 可微；该函数关于 $\boldsymbol{\phi}$可微：

$$\boldsymbol{\theta} \sim q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \quad \Leftrightarrow \quad \boldsymbol{\theta} = T_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}),\ \boldsymbol{\epsilon} \sim p_{base}(\boldsymbol{\epsilon}). \tag{2.23}$$

例如，对于 D维因子化高斯分布，其均值为 $\boldsymbol{\mu} = [\mu_1, \cdots \ \mu_d]^\top$，方差为 $\boldsymbol{\sigma}^2 = [\sigma_1^2, \cdots \sigma_d^2]^\top$， $q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{\theta}_d;\ \mu_d,\ \sigma_d)$，此采样操作可写为 $\boldsymbol{\theta} = T_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})$: $= \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$，其中 $\boldsymbol{\epsilon}_d \sim \mathcal{N}(0,\ 1)$，而 $\odot$表示逐元素乘法。因此，式2.21 中期望对数似然项的蒙特卡洛估计可利用变量替换法则改写如下：

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}[\log p(\mathcal{D}|\boldsymbol{\theta})] \approx \frac{1}{M} \sum_{m=1}^{M} \log p(\mathcal{D}|\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_m), \quad \boldsymbol{\epsilon}_m \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{2.24}$$

此外，对于关于数据点可分解的似然，可以采用基于上述期望对数似然项小批量估计的随机优化方法（如随机梯度下降），从而将变分推断扩展到大型数据集。

变分推断已广泛用于概率模型中的可扩展推断，例如高斯过程和贝叶斯神经网络。接下来，我们将详细回顾其在高斯过程模型中的应用。由于权重空间变分推断相关研究与本学位论文的研究没有直接关系，若不提及贝叶斯神经网络背景下的一些优秀成果，未免有所疏漏。我们向有兴趣的读者推荐Arbel 等人（2023年）对这一主题的综述。

2.3 高斯过程

高斯过程（GP；（Rasmussen 和 Williams，2006年））是一类强大的贝叶斯非参数模型，广泛用于在不确定性下推断未知函数。GP 不假设模型具有参数化形式，也不对参数进行后验推断，而是直接指定函数值的分布。非正式地说，GP 可视为定义在指标集 $\mathcal{X}$（ f的定义域）上的函数值的无限维高斯分布，其中该指标集可以是

无限的，例如 $\mathcal{X} = \mathbb{R}^D$。

高斯过程由均值函数 $m_{\boldsymbol{\varphi}}(\cdot) : \mathcal{X} \to \mathbb{R}$ 以及对称正定的协方差函数完全确定，该协方差函数通常由核函数$k_{\boldsymbol{\varphi}}(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $p(f;\ \boldsymbol{\varphi}) = \mathcal{GP}(f;\ m_{\boldsymbol{\varphi}}(\cdot)\ k_{\boldsymbol{\varphi}}(\cdot,\cdot))$ 参数化。这里，$\boldsymbol{\varphi}$用于表示与均值函数和核函数相关的超参数。

高斯过程通过首先对未知函数 f 设定高斯过程先验，在函数空间中提供了一种强大的贝叶斯建模范式。通常使用无信息先验均值函数，例如零函数 $m(\cdot) := 0$，本学位论文将始终遵循这一惯例。

$$p(f;\boldsymbol{\varphi}) = \mathcal{GP}(f;0,k_{\boldsymbol{\varphi}}(\cdot,\cdot)). \tag{2.25}$$

值得注意的是，作为随机过程，高斯过程的一个核心性质是边缘一致性。例如，根据上述高斯过程先验，在输入位置的有限子集 $\boldsymbol{X} := [\boldsymbol{x}_1, \cdots\ \boldsymbol{x}_N]^\top \in \mathcal{X}$上，函数值 $\boldsymbol{f_X} := [f(\boldsymbol{x}_1), \cdots\ f(\boldsymbol{x}_N)]^\top$的边缘分布是如下多元高斯分布：

$$p(\boldsymbol{f_X};\boldsymbol{\varphi}) = \mathcal{N}(\boldsymbol{f_X};\mathbf{0},\boldsymbol{K_{XX}}), \tag{2.26}$$

其中，协方差矩阵由核矩阵 $\boldsymbol{K_{XX}}$ 得到，其第 ij 个元素是 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的核值 $[\boldsymbol{K_{XX}}]_{ij} = k_{\boldsymbol{\varphi}}(\boldsymbol{x}_i,\ \boldsymbol{x}_j)$。请注意，$\boldsymbol{K_{XX}}$ 取决于超参数 $\boldsymbol{\varphi}$，但为简洁起见，我们在其符号中省略了 $\boldsymbol{\varphi}$ 。

2.3.1 回归的精确推断

对于具有观测数据 $\mathcal{D} = (\boldsymbol{X},\ \boldsymbol{y}) := \{\boldsymbol{x}_n,\ y_n\}_{n=1}^N$ 的回归问题，通常假设采用噪声方差为 σ^2 的独立同分布高斯似然，

$$p(\boldsymbol{y}|\boldsymbol{f_X};\sigma^2) = \mathcal{N}(\boldsymbol{y};\boldsymbol{f_X},\sigma^2\boldsymbol{I}), \tag{2.27}$$

由于高斯先验与似然共轭，后验过程也是一个高斯过程（GP），因此可以得到以下闭式后验预测分布：

$$\begin{aligned} p(\boldsymbol{y}^*|\boldsymbol{X}^*,\mathcal{D};\boldsymbol{\varphi}) = \mathcal{N}(\boldsymbol{y}^*;&\boldsymbol{K_{X^*X}}(\boldsymbol{K_{XX}}+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y},\\ &\boldsymbol{K_{X^*X^*}}+\sigma^2\boldsymbol{I}\underbrace{-\boldsymbol{K_{X^*X}}(\boldsymbol{K_{XX}}+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{K_{XX^*}}}_{\text{uncertainty reduction}}), \end{aligned} \tag{2.28}$$

其中，$\boldsymbol{X}^*$ 为测试输入。后验预测协方差中的最后一项通常称为不确定性降低，其大小取决于测试输入 $\boldsymbol{X}^*$ 与观测输入 $\boldsymbol{X}$之间的相似度，该相似度由核来度量。对于更接近 $\boldsymbol{X}$的 $\boldsymbol{X}^*$，$\boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}}$ 会更大，从而带来更大的不确定性降低。例如，如果使用基于欧几里得距离的核（如径向基函数核），预测不确定性就会考虑欧几里得距离。一般而言，可以指定核来编码关于潜在函数的一些先验假设（例如平滑性、周期性）。

核超参数 $\boldsymbol{\varphi}$（例如径向基函数核中的长度尺度）以及观测噪声方差 σ^2 都会显著影响高斯过程的预测。常见的选择方法是最大化对数边际似然（也称为 II 型最大似然估计）。对于回归问题，对数边际似然也可以写成如下闭式：

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{X};\boldsymbol{\varphi}) &= \log \int p(\boldsymbol{y}|\boldsymbol{f}_{\boldsymbol{X}};\boldsymbol{\varphi})p(\boldsymbol{f}_{\boldsymbol{X}};\boldsymbol{\varphi})d\boldsymbol{f}_{\boldsymbol{X}} \\ &= \log \mathcal{N}(\boldsymbol{y};\boldsymbol{0},\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}+\sigma^2\boldsymbol{I}) \\ &= \underbrace{-\frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}}_{\text{data fit}} \underbrace{-\frac{1}{2}\log\det(\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}+\sigma^2\boldsymbol{I})}_{\text{model complexity}} + \text{const.}\end{aligned} \tag{2.29}$$

它由两项组成，分别权衡数据拟合与模型复杂度惩罚，最终选出能够良好解释数据的最简单模型所对应的超参数配置。

有趣的是，高斯过程的后验推断可以通过核特征映射下的贝叶斯线性回归视角重新表述（Rasmussen 和 Williams，2006年，第 2 章）。考虑如下线性回归问题。

$$y(x) = \boldsymbol{w}^\top\Phi(\boldsymbol{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0,\sigma^2), \tag{2.30}$$

其中，$\Phi(\cdot) : \mathrm{X} \to \mathbb{R}^{d_\Phi}$ (d_Φ 可以是 ∞) 是与核$k(\cdot,\cdot)$ 相关联的特征映射，使得 $k(\boldsymbol{x},\boldsymbol{x}') = \Phi(\boldsymbol{x})^\top\Phi(\boldsymbol{x}')$。Mercer 定理（Mercer，1909）保证了满足 Mercer 条件的核函数存在这样的特征映射。权重上的标准高斯先验 $p(\boldsymbol{w}) = \mathrm{N}(0,\boldsymbol{I})$ 会在函数空间中诱导出一个高斯过程先验：给定数据 $\mathrm{D} = (\boldsymbol{X},\boldsymbol{y})$，我们知道，由于模型的线性性，在 $\boldsymbol{X}$ 上评估的函数值服从高斯分布，其均值和协方差分别为 0 和 $\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}+\sigma^2\boldsymbol{I}$，这恰好

与高斯过程模型中 $\boldsymbol{y}_{\boldsymbol{X}}$ 的先验相匹配。权重的后验分布是一个高斯 n

$$
\begin{aligned}
p(\boldsymbol{w}|\mathcal{D}) &= \mathcal{N}(\boldsymbol{w}; \boldsymbol{m}_{\boldsymbol{w}}, \boldsymbol{S}_{\boldsymbol{w}}), \\
\boldsymbol{m}_{\boldsymbol{w}} &= \frac{1}{\sigma^2}\boldsymbol{S}_{\boldsymbol{w}}\boldsymbol{\Phi}_{\boldsymbol{X}}^{\top}\boldsymbol{y}, \\
\boldsymbol{S}_{\boldsymbol{w}} &= (\boldsymbol{I} + \frac{1}{\sigma^2}\boldsymbol{\Phi}_{\boldsymbol{X}}^{\top}\boldsymbol{\Phi}_{\boldsymbol{X}})^{-1} \\
&= \boldsymbol{I} - \boldsymbol{\Phi}_{\boldsymbol{X}}^{\top}(\sigma^2\boldsymbol{I} + \boldsymbol{\Phi}_{\boldsymbol{X}}\boldsymbol{\Phi}_{\boldsymbol{X}}^{\top})^{-1}\boldsymbol{\Phi}_{\boldsymbol{X}} \qquad \text{(Woodbury identity)} \\
&= \boldsymbol{I} - \boldsymbol{\Phi}_{\boldsymbol{X}}^{\top}(\sigma^2\boldsymbol{I} + \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}})^{-1}\boldsymbol{\Phi}_{\boldsymbol{X}},
\end{aligned}
\tag{2.31}
$$

其中 $\boldsymbol{\Phi}_{\boldsymbol{X}} \in \mathbb{R}^{N\times d_\phi}$是设计矩阵，并注意到 $\boldsymbol{\Phi}_{\boldsymbol{X}}\boldsymbol{\Phi}_{\boldsymbol{X}}^{\top} = \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}$。基于权重后验分布，可以推导出新输入位置 $\boldsymbol{X}^*$ 的后验预测，其为高斯分布，如下所示：

$$
\begin{aligned}
p(\boldsymbol{y}^*|\boldsymbol{X}^*, \mathcal{D}) &= \int p(\boldsymbol{y}^*|\boldsymbol{w}, \boldsymbol{X}^*)p(\boldsymbol{w}|\mathcal{D})d\boldsymbol{w} \\
&= \mathcal{N}(\boldsymbol{y}^*; \boldsymbol{\Phi}_{\boldsymbol{X}^*}\boldsymbol{m}_{\boldsymbol{w}}, \sigma^2\boldsymbol{I} + \boldsymbol{\Phi}_{\boldsymbol{w}}^{\top}\boldsymbol{S}_{\boldsymbol{w}}\boldsymbol{\Phi}_{\boldsymbol{w}}).
\end{aligned}
\tag{2.32}
$$

很容易看出，后验预测协方差与式 2.28中的后验高斯过程完全相同。后验预测均值还可以进一步简化如下：

$$
\begin{aligned}
\boldsymbol{\Phi}_{\boldsymbol{X}^*}\boldsymbol{m}_{\boldsymbol{w}} &= \frac{1}{\sigma^2}\boldsymbol{\Phi}_{\boldsymbol{X}^*}\boldsymbol{S}_{\boldsymbol{w}}\boldsymbol{\Phi}_{\boldsymbol{X}}^{\top}\boldsymbol{y} \\
&= \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}}\left\{\frac{1}{\sigma^2}\left[\boldsymbol{I} - (\sigma^2\boldsymbol{I} + \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}})^{-1}\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}\right]\right\}\boldsymbol{y} \\
&= \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}}(\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}.
\end{aligned}
\tag{2.33}
$$

对于最后一行，只需注意到

$$
(\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} + \sigma^2\boldsymbol{I})\left\{\frac{1}{\sigma^2}\left[\boldsymbol{I} - (\sigma^2\boldsymbol{I} + \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}})^{-1}\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}\right]\right\} = \frac{1}{\sigma^2}\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} + \boldsymbol{I} - \frac{1}{\sigma^2}\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} = \boldsymbol{I}.
$$

因此，基于核特征映射的高斯过程模型等价于贝叶斯线性回归模型。

2.3.2 稀疏变分高斯过程

尽管高斯过程在小数据情形下表现出色，并能提供可靠的预测不确定性，但其后验推断需要对观测数据点数量为 $O(N^3)$ 的矩阵求逆，计算代价呈三次方增长，因此在大规模数据集上计算上难以实现。将高斯过程拓展到大数据应用的一种方法，是在变分推断框架下使用具有低秩结构的近似后验高斯过程。一种广受欢迎的方法是稀疏

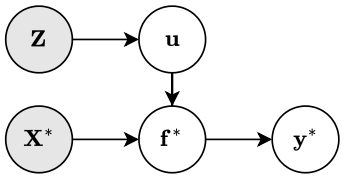


图 2.3：稀疏变分高斯过程（SVGP）预测的图模型。

变分高斯过程（SVGP；（Snelson 和 Ghahramani，2005）），它使用高斯过程 Q 来近似后验过程；该高斯过程基于一小组增广诱导点（伪输入—函数值对）构建，其数量为 M（通常为 $M \ll N$），$\{(\boldsymbol{z}_m, u_m)\}_{m=1}^{M}$（Hensman 等人，2013年、2015年）。SVGP 将完整高斯过程的 $O(N^3)$ 计算成本降低至 $O(M^3 + NM^2)$；这是因为根据构造，Q完全由这一小组诱导点的近似后验决定，而该近似后验被设定为一个高斯分布，其具有可学习的变分均值 $\boldsymbol{m_u}$ 和变分协方差 $\boldsymbol{S_u}$，$q(\boldsymbol{u}) = \mathrm{N}(\boldsymbol{m_u}, \boldsymbol{S_u})$。

根据标准高斯过程先验，$p(f) =$ 高斯过程$(0(\cdot)\ k_{\varphi}(\cdot,\cdot))$，由 $\boldsymbol{u}$ 和 $\boldsymbol{f}^*$组成的增广集合在任意输入位置集合 $\boldsymbol{X}^*$处的先验边缘分布为

$$p(\boldsymbol{f}^*, \boldsymbol{u}|\boldsymbol{X}^*, \boldsymbol{Z}) = \mathcal{N}(0, \begin{pmatrix} \boldsymbol{K_{X^*X^*}} & \boldsymbol{K_{X^*Z}} \\ \boldsymbol{K_{ZX^*}} & \boldsymbol{K_{ZZ}}. \end{pmatrix}). \tag{2.34}$$

上述计算增益是通过对近似后验高斯过程施加以下低秩结构实现的。给定诱导点的近似后验 $q(\boldsymbol{u})$，假设给定 $\boldsymbol{u}$ 时$\boldsymbol{f}^*$ 的近似后验条件分布与先验条件分布相同。

$$q(\boldsymbol{f}^*|\boldsymbol{u}, \boldsymbol{Z}, \boldsymbol{X^*}) = p(\boldsymbol{f}^*|\boldsymbol{u}, \boldsymbol{Z}, \boldsymbol{X^*}). \tag{2.35}$$

因此，$\boldsymbol{f}^*$ 的诱导近似后验是一个高斯分布，其低秩结构由诱导点定义：

$$\begin{aligned} q(\boldsymbol{f}^*|\boldsymbol{X}^*, \boldsymbol{Z}) &= \int p(\boldsymbol{f}^*|\boldsymbol{u}, \boldsymbol{Z}, \boldsymbol{X^*}) q(\boldsymbol{u}) d\boldsymbol{u} \\ &= \mathcal{N}(\boldsymbol{f}^*; \boldsymbol{K_{X^*Z}} \boldsymbol{K_{ZZ}^{-1}} \boldsymbol{m_u}, \boldsymbol{K_{X^*X^*}} + \boldsymbol{K_{X^*Z}} \boldsymbol{K_{ZZ}^{-1}} (\boldsymbol{S_u} - \boldsymbol{K_{ZZ}}) \boldsymbol{K_{ZZ}^{-1}} \boldsymbol{K_{ZX^*}}). \end{aligned} \tag{2.36}$$

请注意，现在矩阵求逆的对象是 $\boldsymbol{K_{ZZ}}$，其 $O(M^3)$计算成本显著低于完整高斯过程中 $O(N^3)$求逆的成本。图 2.3 展示了稀疏变分高斯过程（SVGP）预测的图模型。此外，从2.3

从权重空间推断的角度来看，SVGP 的预测等价于基于核特征映射的近似贝叶斯线性模型，$f(x) = \boldsymbol{w}^\top \Phi(\boldsymbol{x})$，其中考虑了由变分参数 $\boldsymbol{m_u}$ 和 $\boldsymbol{S_u}$参数化的、关于 $\boldsymbol{w}$ 的以下结构化低秩近似后验。

$$q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\Phi_Z^\top K_{ZZ}^{-1} m_u}, \boldsymbol{I} + \boldsymbol{\Phi_Z^\top K_{ZZ}^{-1}}(\boldsymbol{S_u} - \boldsymbol{K_{ZZ}})\boldsymbol{K_{ZZ}^{-1}\Phi_Z}). \tag{2.37}$$

给定观测数据 $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$，诱导输入位置 $\boldsymbol{Z}$ 被视为模型超参数，并通过证据下界最大化与核超参数及变分参数共同调优。

$$\begin{aligned}
\mathcal{L}_{ELBO} &= \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u},|\boldsymbol{Z},\boldsymbol{X})}[\log \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X})p(\boldsymbol{u}|\boldsymbol{Z})}{q(\boldsymbol{f},\boldsymbol{u}|\boldsymbol{Z},\boldsymbol{X})}] \\
&= \int\int p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X})q(\boldsymbol{u}) \log \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{u}|\boldsymbol{Z})}{q(\boldsymbol{u})} d\boldsymbol{u} d\boldsymbol{f} \\
&= \underbrace{\mathbb{E}_{q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{Z})}[\log p(\boldsymbol{y}|\boldsymbol{f})]}_{\text{ELL}} - \underbrace{\mathrm{KL}(q(\boldsymbol{u})||p(\boldsymbol{u}|\boldsymbol{Z}))}_{\text{KL regularizer}}
\end{aligned} \tag{2.38}$$

上述证据下界的完整推导见附录 B.1.1。变分推断框架也允许使用非高斯观测似然，从而将高斯过程的应用范围拓展到回归任务之外。对于非高斯似然，期望对数似然项没有闭式解，因此通常采用蒙特卡洛估计，其主要计算成本（$O(M^3 + NM^2)$）来自计算 $q(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{Z})$。由于 $q(\boldsymbol{u})$ 和 $p(\boldsymbol{u}|\boldsymbol{Z})$ 均为高斯分布，因此 KL 正则项可以解析计算，其主要计算成本来自计算行列式 $\det \boldsymbol{K_{ZZ}}$以及矩阵求逆 $\boldsymbol{K_{ZZ}^{-1}}$（两者均为 $O(M^3)$）。综合而言，计算$\mathcal{L}_{ELBO}$时，稀疏变分高斯过程的计算成本为 $O(M^3 + NM^2)$。

$\boldsymbol{y}^*$ 在测试输入位置 $\boldsymbol{X}^*$ 处的后验预测分布为

$$\begin{aligned}
p(\boldsymbol{y}^*|\boldsymbol{X}^*, \boldsymbol{Z}, \boldsymbol{u}) &= \int\int p(\boldsymbol{y}^*|\boldsymbol{f}^*)p(\boldsymbol{f}^*|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X^*})q(\boldsymbol{u})d\boldsymbol{u}d\boldsymbol{f}^* \\
&= \mathbb{E}_{q(\boldsymbol{f}^*|\boldsymbol{X}^*,\boldsymbol{Z})}[p(\boldsymbol{y^*}|\boldsymbol{f^*})].
\end{aligned} \tag{2.39}$$

同样，对于需要非高斯观测似然的任务，我们可以采用蒙特卡洛方法对其进行估计。

对于基于高斯观测似然、噪声方差为 σ^2的回归，期望对数似然项也可以解析计算。因此，最优变分分布 $q_*(\boldsymbol{u})$可以解析推导，且它是一个高斯分布，

$$q_*(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \frac{1}{\sigma^2}\boldsymbol{K_{ZZ}M^{-1}K_{ZX}y}, \boldsymbol{K_{ZZ}M^{-1}K_{ZZ}}), \tag{2.40}$$

其中 $\boldsymbol{M}:=\boldsymbol{K}_{\boldsymbol{ZZ}}+\frac{1}{\sigma^2}\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{k}_{\boldsymbol{XZ}}$。在这种情况下，我们通常将 SVGP 称为稀疏高斯过程回归（SGPR；（Titsias，2009年））。将 $q_*(\boldsymbol{u})$ 代回证据下界，可得到所谓的折叠变分界，它将成为进一步调整核超参数 $\boldsymbol{\varphi}$ 和诱导输入位置$\boldsymbol{Z}$的目标函数，

$$\begin{aligned}\mathcal{L}_{ELBO}(q_*)=\log\mathcal{N}(\boldsymbol{y};\mathbf{0},&\sigma^2\boldsymbol{I}+\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{ZZ}}^{-\mathbf{1}}\boldsymbol{K}_{\boldsymbol{ZX}})\\&-\frac{1}{2\sigma^2}\mathrm{Tr}(\boldsymbol{K}_{\boldsymbol{XX}}-\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{K}_{\boldsymbol{ZZ}}^{-\mathbf{1}}\boldsymbol{K}_{\boldsymbol{ZX}}).\end{aligned}\tag{2.41}$$

自然，相应的后验预测分布也具有解析可处理的高斯形式。

$$\begin{aligned}p(\boldsymbol{y}^*|\boldsymbol{X}^*,\mathcal{D},\boldsymbol{Z})=\mathcal{N}(\boldsymbol{y}^*;&\frac{1}{\sigma^2}\boldsymbol{K}_{\boldsymbol{X^*Z}}\boldsymbol{M}^{-\mathbf{1}}\boldsymbol{K}_{\boldsymbol{ZX}}\boldsymbol{y},\\&\boldsymbol{K}_{\boldsymbol{X^*X^*}}+\sigma^2\boldsymbol{I}\underbrace{-\boldsymbol{K}_{\boldsymbol{X^*Z}}(\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}-\boldsymbol{M}^{-1})^{-1}\boldsymbol{K}_{\boldsymbol{ZX^*}}}_{\text{uncertainty reduction}}).\end{aligned}\tag{2.42}$$

对于稀疏变分高斯过程，诱导点可以视为训练集的紧凑摘要，而后验预测中的不确定性降低取决于 $\boldsymbol{X}^*$ 与 $\boldsymbol{Z}$ 之间的核相似度（这往往覆盖训练输入的整个区域）。

2.3.3 深度核学习与深度高斯过程

与深度神经网络相比，高斯过程在复杂预测任务中的表现可能不尽如人意，这并不奇怪。毕竟，正如前文所述，它们只是基于核特征的浅层贝叶斯线性模型。本节将介绍两种利用深层结构增强高斯过程的常见方法。

深度核学习。线性模型依赖人工设计的特征映射，而如果将网络最后一层之前的部分视为表达能力强的非线性特征提取器，那么深度神经网络便能够为输入数据学习到丰富的特征表示。因此，改进高斯过程的一种自然方法是结合核特征映射与深度特征提取器。深度核学习（Wilson 等人，2016年）通过使用深度神经网络对核函数进行参数化，引入了深度核。网络权重 $\boldsymbol{\theta}$ 由此成为深度核的超参数。具体而言，深度核由一个常规基核 $k_{base}(\cdot,\cdot)$（例如径向基函数核）与一个由深度神经网络参数化的深度特征提取器复合而成（$h_{\boldsymbol{\theta}}(\cdot)$）：$k_{deep}(\cdot,\cdot)=k_{base}(h_{\boldsymbol{\theta}}(\cdot)\ h_{\boldsymbol{\theta}}(\cdot))$。

深度高斯过程。除了在核函数中融入深度结构外，多个高斯过程层本身也可以堆叠起来，构成所谓的深度高斯过程（deep GPs）（Damianou 和 Lawrence，2013）。与标准高斯过程不同，深度高斯过程不是线性模型，因此后验推断难以处理。我们再次采用前文讨论的变分诱导点方法，近似每个高斯过程层的后验过程（Salimbeni 和 Deisenroth，2017）。考虑一个由 $f_{l=1}^{L}$ 个潜在函数组成的 L 层深度高斯过程。每一层的输入 f_l 是前一层输出的函数值： $\boldsymbol{f}_l = f_l(\boldsymbol{f}_{l-1})$。每一层均使用高斯过程先验来描述其输出： $p(f_l) = \mathrm{GP}(0(\cdot)\ k^{(l)}(\cdot,\cdot))$。此外，我们还为每一层引入一组诱导点 $\{\boldsymbol{z}_l^{(i)}, u_l^{(i)}\}_{i=1}^{M_l}$。先验模型于是为

$$p(\boldsymbol{y}, \{\boldsymbol{f}_l\}_{l=1}^{L}, \{\boldsymbol{u}\}_{l=1}^{L} | \boldsymbol{X}, \{\boldsymbol{Z}_l\}_{l=1}^{L}) = p(\boldsymbol{y}|\boldsymbol{f}_L)\left[\prod_{l=2}^{L} p(\boldsymbol{f}_l|\boldsymbol{u}_l, \boldsymbol{Z}_l, \boldsymbol{f}_{l-1}) p(\boldsymbol{u}_l|\boldsymbol{Z}_l)\right] p(\boldsymbol{f}_1|\boldsymbol{u}_1, \boldsymbol{Z}_1, \boldsymbol{X}) p(\boldsymbol{u}_1|\boldsymbol{Z}_1). \tag{2.43}$$

假设 $\{\boldsymbol{u}_l\}_{l=1^L}$ 的近似后验在各层之间是因子化的，并且在每一层中，给定 $\boldsymbol{u}_l$ 时 $\boldsymbol{f}_l$ 的近似后验条件分布再次假设与先验条件分布相同。总之，联合近似后验因子化如下：

$$q(\{\boldsymbol{f}_l\}_{l=1}^{L}, \{\boldsymbol{u}_l\}_{l=1}^{L} | \boldsymbol{X}, \{\boldsymbol{Z}_l\}_{l=1}^{L}) = \left[\prod_{l=2}^{L} p(\boldsymbol{f}_l|\boldsymbol{u}_l, \boldsymbol{Z}_l, \boldsymbol{f}_{l-1}) q(\boldsymbol{u}_l)\right] p(\boldsymbol{f}_1|\boldsymbol{u}_1, \boldsymbol{Z}_1, \boldsymbol{X}) q(\boldsymbol{u}_1). \tag{2.44}$$

上述结构化近似后验使我们能够得到以下证据下界，作为训练目标。

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q(\boldsymbol{f}_L|\boldsymbol{X}, \{\boldsymbol{Z}_l\}_{l=1}^{L})}[\log p(\boldsymbol{y}|\boldsymbol{f}_L)] - \sum_{l=1}^{L} \mathrm{KL}(q(\boldsymbol{u}_l)||p(\boldsymbol{u}_l|\boldsymbol{Z}_l)), \tag{2.45}$$

其中

$$q(\boldsymbol{f}_L|\boldsymbol{X}, \{\boldsymbol{Z}_l\}_{l=1}^{L}) = \int \cdots \int q(\boldsymbol{f}_1|\boldsymbol{X}, \boldsymbol{Z}_1) \prod_{l=2}^{L} q(\boldsymbol{f}_l|\boldsymbol{f}_{l-1}, \boldsymbol{Z}_l) d\boldsymbol{f}_{1:L-1}. \tag{2.46}$$

由上述积分定义的最终输出的近似后验分布无法进行解析求解，实际中我们只能使用从中采样得到的样本。通常，通过将从 $q(\boldsymbol{f}_l|\boldsymbol{f}_{l-1}, \boldsymbol{Z}_l)$ 中采样得到的样本依次传递通过每一层，即可得到最终输出 $q(\boldsymbol{f}_L|\boldsymbol{x}, \{\boldsymbol{Z}_l\}_{l=1}^{L})$，而这些样本可以

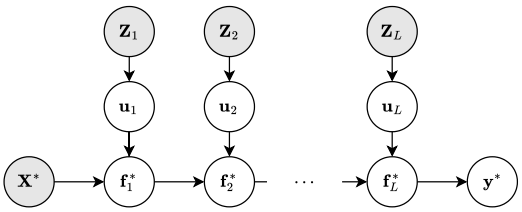


图 2.4：具有稀疏高斯过程近似的 L 层深度高斯过程的预测图模型。

利用重参数化技巧生成，因为 $q(\boldsymbol{f}_l|\boldsymbol{f}_{l-1},\ \boldsymbol{Z}_l)$ 服从高斯分布，

$$\begin{aligned} q(\boldsymbol{f}_l|\boldsymbol{f}_{l-1},\boldsymbol{Z}_l) &= \int p(\boldsymbol{f}_l|\boldsymbol{u}_l,\boldsymbol{Z}_l,\boldsymbol{f}_{l-1})q(\boldsymbol{u}_l)d\boldsymbol{u}_l \\ &= \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f}_{l-1}\boldsymbol{Z}_l}^{(l)}\boldsymbol{K}_{\boldsymbol{Z}_l\boldsymbol{Z}_l}^{(l)-\mathbf{1}}\boldsymbol{m}_{\boldsymbol{u}_l}, \\ &\quad \boldsymbol{K}_{\boldsymbol{f}_{l-1}\boldsymbol{f}_{l-1}}^{(l)} + \boldsymbol{K}_{\boldsymbol{f}_{l-1}\boldsymbol{Z}_l}^{(l)}\boldsymbol{K}_{\boldsymbol{Z}_l\boldsymbol{Z}_l}^{(l)-\mathbf{1}}(\boldsymbol{S}_{\boldsymbol{u}_l} - \boldsymbol{K}_{\boldsymbol{Z}_l\boldsymbol{Z}_l}^{(l)})\boldsymbol{K}_{\boldsymbol{Z}_l\boldsymbol{Z}_l}^{(l)-\mathbf{1}}\boldsymbol{K}_{\boldsymbol{Z}_l\boldsymbol{f}_{l-1}}^{(l)}). \end{aligned} \tag{2.47}$$

我们在图中展示了深度高斯过程预测的图模型2.4。

2.4 深度序列生成模型

除了预测模型之外，我们还利用概率机器学习和深度神经网络来构建深度生成模型。由于这两个领域都取得了巨大进展，深度生成模型如今已成为一个快速发展的研究领域。简而言之，深度生成模型旨在利用从由$p(\boldsymbol{x})$抽取的样本组成的训练集学习得到的参数化模型 $p_{\boldsymbol{\theta}}(\boldsymbol{x})$，来估计真实的数据分布 $p(\boldsymbol{x})$，$\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$，其中 $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ 由参数为 $\boldsymbol{\theta}$ 的深度神经网络进行参数化。学习完成后，可以从 $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ 生成新样本，这些样本可视为来自真实数据生成过程的近似样本。本节介绍深度序列生成模型的框架，并详细讨论该框架中的两种流行模型：层次变分自编码器和扩散模型。

2.4.1 深度潜变量模型

我们首先回顾深度潜变量模型（DLVMs；（Kingma 和 Welling，2014年；Rezende 等人，2014年））， 因为许多深度序列生成模型都可以视为深度潜变量模型的扩展。深度潜变量模型假设，每个观测 $\boldsymbol{x}$ 都由潜变量 $\boldsymbol{z}$经由参数为$\boldsymbol{\theta}(\cdot)$ 的深度神经网络 $f_{\boldsymbol{\theta}}$进行变换而生成。通常，潜变量的分布 $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ 先验地选为一种简单分布（例如标准高斯分布），以便于从中采样。图 2.5 展示了深度潜变量模型的图模型。从数学上说，

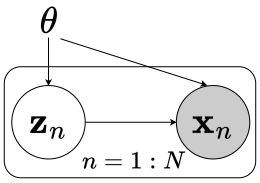


图 2.5：深度潜变量模型的图模型。

生成模型可以写成如下形式。

$$\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z}) := \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) := \mathcal{N}(f_{\boldsymbol{\theta}}(\boldsymbol{z}), \sigma^2 \boldsymbol{I}). \tag{2.48}$$

给定训练集 $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$，理想情况下，可以通过最大似然估计（MLE）拟合模型参数 $\boldsymbol{\theta}$：

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_n), \quad p_{\boldsymbol{\theta}}(\boldsymbol{x}_n) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z}. \tag{2.49}$$

然而，上述 MLE 目标函数（式 2.49）需要对潜变量 $\boldsymbol{z}_n$ 求解复杂积分，以计算对数边际似然 $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$，但由于 $f_{\boldsymbol{\theta}}(\boldsymbol{z})$ 是一个非线性深度神经网络，这些积分难以计算。实践中，我们再次采用近似推断，许多深度潜变量模型通过基于变分推断的方法进行学习（Kingma 和 Welling，2014年；Rezende 等人，2014年）。我们将详细讨论一类具有代表性的基于变分推断的深度潜变量模型——变分自编码器。

变分自编码器。变分自编码器（VAEs）（Kingma 和 Welling，2014年；Rezende 等人，2014年）是一类利用 VI 进行近似最大似然最大化的 DLVMs。值得注意的是，为了降低计算成本，它们进一步采用摊销推断（Gershman 和 Goodman，2014年）来构建 DLVMs 中的近似分布：VAEs 不再针对每个 $\boldsymbol{x}_n$ 单独构建后验分布 $p_{\boldsymbol{\theta}}(\boldsymbol{z}_n|\boldsymbol{x}_n)$ 的近似分布，而是使用另一个深度神经网络（通常称为推断网络或编码器网络），将每个 $\boldsymbol{x}_n$ 映射到其对应变分分布的变分参数。例如，当考虑因子化高斯分布族 $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x}))$ 时，与观测 $\boldsymbol{x}$ 相关的变分均值和方差随后由推断网络 $g_{\boldsymbol{\phi}}(\boldsymbol{x}) := [\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \log \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})]$ 获得，其中使用了摊销变分参数 $\boldsymbol{\phi}$。对于每个观测 $\boldsymbol{x}_n$，都可以将证据下界构造为相应对数边际似然的下界，并将这些证据下界求和，以形成用于联合训练模型参数的最终目标函数

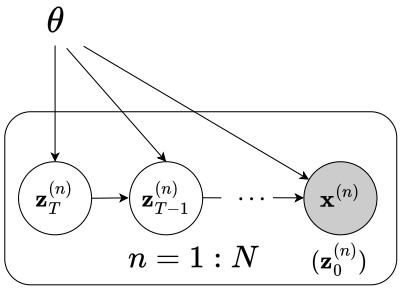


图 2.6：深度序列生成变量模型的图模型 .

参数 $\boldsymbol{\theta}$ 和变分参数 $\boldsymbol{\phi}$:

$$\begin{aligned} \boldsymbol{\theta}^*, \boldsymbol{\phi}^* &= \arg\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \sum_{n=1}^{N} \mathcal{L}_{ELBO}(\boldsymbol{x}_n, \boldsymbol{\theta}, q_{\boldsymbol{\phi}}), \\ \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \mathcal{L}_{ELBO}(\boldsymbol{x}, \boldsymbol{\theta}, q_{\boldsymbol{\phi}}) &= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})]}_{\text{ELL}} - \underbrace{\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\boldsymbol{z}))}_{\text{KL regularizer}}. \end{aligned} \tag{2.50}$$

期望对数似然项中的棘手期望同样可以通过蒙特卡洛估计进行近似，通常只需一个样本就足以取得不错的实证性能。为使推断网络能够进行反向传播，再次采用重参数化技巧（Kingma 和 Welling，2014年；Rezende 等人，2014年）从 $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$中采样:

$$\mathcal{L}_{ELBO}(\boldsymbol{x}, \boldsymbol{\theta}, q_{\boldsymbol{\phi}}) \approx \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}) \odot \boldsymbol{\epsilon}) - \mathrm{KL}(q_{\boldsymbol{\phi}}\|p_{\boldsymbol{\theta}}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{2.51}$$

推断网络 $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ 和 DLVM 的条件分布 $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ 分别可以视为随机编码器和解码器。这也解释了为什么我们将该方法称为“自编码器”。

2.4.2 层次变分自编码器

对于仅有一个潜变量的深度潜变量模型（DLVM），生成逼真的观测（例如照片级真实感图像）可能颇具挑战，因为它们试图一步将噪声转换为目标分布。深度序列生成模型（DSGM）通过引入一系列潜变量来扩展深度潜变量模型，并利用多个中间步骤将噪声转换为目标分布。具体而言，它们采用一系列潜变量 $\boldsymbol{z}_1, \cdots \boldsymbol{z}_T$ 来捕捉数据 $\boldsymbol{z}_0$ 的表示：$= \boldsymbol{x}$ 在不同的保真度下（Salimans，2016 年）：

$$\boldsymbol{x} = \boldsymbol{z}_0 \sim p_{\boldsymbol{\theta}}(\boldsymbol{z}_0) = \int p(\boldsymbol{z}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) d\boldsymbol{z}_{1:T}, \tag{2.52}$$

其中，最后一个潜变量 $\boldsymbol{z}_T$ 的先验分布通常设为一种简单分布，例如标准高斯分布 $p(\boldsymbol{z}_T) := \mathcal{N}(\boldsymbol{z}_T; \boldsymbol{0}, \boldsymbol{I})$，而潜条件分布则由深度神经网络参数化（我们通常只对条件均值进行参数化）：

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) := \mathcal{N}(\boldsymbol{z}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t), \sigma_t^2 \boldsymbol{I}). \tag{2.53}$$

我们在图2.6中展示了深度序列生成模型的图模型。

与深度潜变量模型类似，直接对深度序列生成模型的模型参数进行最大似然估计 $\boldsymbol{\theta}$ 是难以处理的，因此，变分推断同样提供了一个有原则的框架，用于构造对数似然的下界目标函数。一种重要的基于变分推断的深度序列生成模型是层次变分自编码器（HVAE；（Sønderby 等人，2016年；Maaløe 等人，2019年；Vahdat 和 Kautz，2020年））。具体而言，层次变分自编码器使用以下分布近似难解后验分布

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0) = \frac{p(\boldsymbol{z}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)}{p_{\boldsymbol{\theta}}(\boldsymbol{z}_0)} \tag{2.54}$$

：摊销变分分布（或称推断模型） $q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0)$。针对推断模型的因子分解方式，人们提出了多种不同的设计选择，包括“自底向上”因子分解（Burda 等人，2015年）：

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0) = \prod_{t=1}^{T} q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}), \tag{2.55}$$

以及“自顶向下”因子分解（Sønderby 等人，2016年）：

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_T|\boldsymbol{z}_0)\prod_{t=1}^{T} q_{\boldsymbol{\phi}}(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{z}_0), \tag{2.56}$$

其中，因子 $q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$ 和 $q_{\boldsymbol{\phi}}(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{z}_0)$ 是因子化高斯分布，其均值和对角方差由神经网络参数化。给定一组训练观测 $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^N$ ($\boldsymbol{z}_0^{(n)} := \boldsymbol{x}^{(n)}$)，我们再次通过最大化证据下界之和，联合训练生成模型和推断模型：

$$\begin{aligned}
\boldsymbol{\theta}^*, \boldsymbol{\phi}^* &= \arg\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \sum_{n=1}^{N} \mathcal{L}_{ELBO}(\boldsymbol{x}^{(n)}, \boldsymbol{\theta}, q_{\boldsymbol{\phi}}), \\
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(n)}) \geq \mathcal{L}_{ELBO}(\boldsymbol{x}^{(n)}, \boldsymbol{\theta}, q_{\boldsymbol{\phi}}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}^{(n)}|\boldsymbol{z}_0^{(n)})}\left[\log \frac{p(\boldsymbol{z}_T^{(n)})\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}^{(n)}|\boldsymbol{z}_t^{(n)})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}^{(n)}|\boldsymbol{z}_0^{(n)})}\right].
\end{aligned} \tag{2.57}$$

还可以再次采用蒙特卡洛估计和重参数化技巧，进一步简化上述目标函数并实现反向传播。

2.4.3 扩散模型

扩散模型（Sohl-Dickstein 等人，2015年；Ho 等人，2020年；Song 等人，2021b）可以说是近年来最成功的深度序列生成模型，并实现了照片级的图像生成质量。它假设一个基于潜变量序列的马尔可夫数据生成过程（或去噪过程），如式 2.52 所示，且条件分布的参数化方式与式2.53类似。不过，与分层变分自编码器不同，扩散模型定义了一个固定的“自下而上”推断模型（或扩散过程）：

$$q(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0) = \prod_{t=1}^{T} q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{z}_t; \sqrt{\alpha_t}\boldsymbol{z}_{t-1}, (1-\alpha_t)\boldsymbol{I}), \quad \boldsymbol{z}_0 := \boldsymbol{x} \tag{2.58}$$

其中，$q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$’ 是预先定义的高斯卷积核，并且不会对潜变量进行降维（即 $d = d_x$，其中 $d_x = \dim(\boldsymbol{x})$）。扩散模型也通过最大化证据下界进行训练，但仅针对生成模型的参数 $\boldsymbol{\theta}$，因为推断模型是固定的，因此不包含任何可训练参数。由于推断模型仅被定义为一系列固定的高斯卷积，因此可以解析地推导出该推断模型对应的“自上而下”版本，其形式如下：

$$q(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{z}_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(\boldsymbol{z}_t, \boldsymbol{z}_0), \tilde{\beta}_t^2), \tag{2.59}$$

˜其中 $\tilde{\mu}$ 和 β_t 都有解析解，且具有闭式表达式。由于推断模型简单且固定，扩散模型针对每个训练观测 $\boldsymbol{x}(\boldsymbol{z}_0)$ 的证据下界可简化为以下形式：

$$\begin{aligned}
\mathcal{L}_{ELBO}(\boldsymbol{x}, \boldsymbol{\theta}) &= \mathbb{E}_{q(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_0|\boldsymbol{z}_1) - \sum_{t=1}^{T} \mathrm{KL}(q(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{z}_0) || p_{\boldsymbol{\theta}}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})) \right] \\
&= \mathbb{E}_{q(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_0|\boldsymbol{z}_1) - \sum_{t=1}^{T} \underbrace{\frac{\|\mu_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t) - \tilde{\mu}(\boldsymbol{z}_t, \boldsymbol{z}_0)\|^2}{2\sigma_t^2}}_{\text{mean matching}} \right].
\end{aligned} \tag{2.60}$$

固定的推断模型为 DSGM 施加了强烈的归纳偏置，因为 DSGM 现在是针对固定扩散过程的去噪过程进行优化的。这种归纳偏置还可以从上述证据下界中“均值匹配”项对每个中间潜变量提供的直接监督中看出。在实践中，在每次训练迭代期间，我们通常只为每个训练观测采样上述证据下界中的一项（由 $T+1$ 项组成），从而构成一个更简单的目标函数。

基于 $\tilde{\mu}$（$\boldsymbol{z}_t$、 $\boldsymbol{z}_0$）的结构，我们也可以相应地对 $\mu_{\boldsymbol{\theta}}$（$\boldsymbol{z}_t$、 t）进行重新参数化，以便将“均值匹配”项改写为其他形式。例如，使用重参数化技巧，扩散模型的噪声参数化（Ho 等人，2020年；Song 等人，2021a）将 $\boldsymbol{z}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\boldsymbol{z}_t - \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\boldsymbol{\epsilon}$ 改写为 $\boldsymbol{\epsilon} \sim \mathrm{N}(0、 \boldsymbol{I})$，其中 $\sqrt{\bar{\alpha}_t}$ 是由固定推断模型确定的固定标量。借助这一点，我们可以进一步将 $\mu_{\boldsymbol{\theta}}$（$\boldsymbol{z}_t$、 t）重新参数化为 $= \tilde{\mu}(\boldsymbol{z}_t、 \frac{1}{\sqrt{\bar{\alpha}_t}}\boldsymbol{z}_t - \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\epsilon_{\boldsymbol{\theta}}(\boldsymbol{z}_t、 t))$，从而将“均值匹配”项改写为“噪声匹配”项：

$$\lambda_t \|\epsilon_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t) - \boldsymbol{\epsilon}\|^2 = \bar{\lambda}_t \|\epsilon_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) - \boldsymbol{\epsilon}\|^2, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{2.61}$$

其中 λ_t 是由 $\tilde{\mu}$ 的解析形式确定的正标量。该目标函数鼓励预测噪声 $\epsilon_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t)$ 与用于将观测 $\boldsymbol{\epsilon}$ 加噪、使其成为加噪版本 $\boldsymbol{x}$ $(\boldsymbol{z}_0)$ 的噪声 $\boldsymbol{z}_t$相匹配。或者，利用去噪均值与加噪模型得分之间的联系，也可以得到基于得分匹配的目标函数来训练扩散模型（Song 等人，2021b）。

第3章

通过稀疏高斯过程校准 Transformer

本章基于 Chen and Li (2023年):

- Wenlong Chen和Yingzhen Li。**通过稀疏高斯过程校准 Transformer**。发表于国际学习表征会议（*ICLR*），2023年。主要想法由我提出，最后一位作者提供了有益的建议。文稿主要由我撰写，最后一位作者协助润色。此外，我编写了全部代码，并完成本工作的所有实验。

本章提出一种近似贝叶斯推断方法——稀疏高斯过程注意力，以提升 Transformer 模型的不确定性量化能力。通用的权重空间推断技术将所有深度神经网络都视为某种参数化模型，并忽略其架构中的归纳偏置。此外，由于前向传播过程中权重之间存在复杂的相互作用，这些权重空间方法没有充分理由偏好任何特定的先验或近似技术。相比之下，稀疏高斯过程注意力的灵感来自于这样一种联系：Transformer 中的注意力架构与稀疏变分高斯过程的后验均值之间存在关联。这种联系自然启发我们在注意力输出上使用高斯过程先验，并基于稀疏高斯过程为其构造近似后验。

3.1 引言

深度序列模型在预测任务上已取得显著进展，正如第 2.1.1 节所述，Transformer（Vaswani 等人，2017年）是一种基于多头注意力（MHA）的深度序列模型架构，近年来广受欢迎。随着 Transformer 被部署于众多下游应用中（Vaswani 等人，2017年；Dosovitskiy 等人，2021年；Brown 等人，2020年），防止模型因产生反复无常且置信度很高的输出而表现出较差的鲁棒性至关重要（Guo 等人，2017；Mukhoti 等人，2020）。这要求对 Transformer 进行校准良好的不确定性量化，而在本研究开展时，这方面的研究还很少。这也引发了人们对将 Transformer 用于安全关键任务的担忧，因为此类任务要求在不确定性下进行理性且审慎规避风险的决策。

在不确定性量化方面，贝叶斯推断是一种强大且有原则的框架，可用于构建概率模型，以便在不确定性下进行合理预测和决策（Gal，2016）。将（近似）贝叶斯推断方法应用于全连接、卷积和循环神经网络的不确定性量化，已取得显著进展（Blundell 等，2015；Gal 和 Ghahramani，2016；Zhang 等，2019；Ritter 等，2021）。研究人员已初步尝试将这些技术扩展到 Transformer，但结果不一（Tran 等，2019；Xue 等，2021）。另一方面，第 2.3 节介绍的高斯过程（GP）是需要可靠函数空间不确定性估计的任务中的黄金标准方法（Rasmussen 和 Williams，2006；Wilson 等，2020）。研究人员提出将深度学习理念融入高斯过程模型设计，包括深度核学习（Wilson 等，2016）和深度高斯过程（Damianou 和 Lawrence，2013；Salimbeni 和 Deisenroth，2017）。然而，这些模型仍未能扩展到现代深度学习任务，例如大规模图像分类和语言建模。

在这项工作中，我们提出了稀疏高斯过程注意力（SGPA），这是一种专为基于注意力的模型（如 Transformer）量身打造的新型不确-定性量化技术，它借鉴了稀疏变分高斯过程（SVGP）（Snelson 和 Ghahramani，2005；Hensman 等人，2013年）中的技术，以改进不确定性估计。我们的工作提出了以下见解并作出以下贡献：

- （第 3.2 节）我们的关键观察是，基于核的注意力（Tsai 等人，2019年）等价于稀疏变分高斯过程的后验均值。这启发我们为注意力输出指定高斯过程先验，并利用稀疏变分高斯过程技术对 Transformer 进行近似后验推断。基于我们的 SGPA 方法得到的 Transformer 可视为一种稀疏深度高斯过程（Salimbeni 和 Deisenroth，2017年），其中每个高斯过程层都使用了深度核。

- （第 3.3 和 3.4 节）我们采用解耦诱导点技术（Cheng 和 Boots，2017年；Salimbeni 等人，2018年），解决将稀疏变分高斯过程直接扩展至多头自注意力时的计算效率低下问题，使 SPGA 能够扩展到 Transformer 所应用的深度学习任务。

- （第 3.5 节）从实证结果来看，在多种视觉、自然语言处理和图预测任务中，与基线方法相比，基于 SGPA 的 Transformer 在分布内校准、分布外（OOD）鲁棒性和 OOD 检测方面均有显著提升，同时准确率也可与采用标准注意力（Vaswani 等人，2017年）或核注意力（Tsai 等人，2019年）的 Transformer 相媲美。

3.2 稀疏高斯过程注意力

我们提出稀疏高斯过程注意力（SGPA），用于对基于 Transformer 的模型执行近似贝叶斯推断。其核心思想是用核（Vaswani 等人，2017年）替换缩放点积注意力中的 softmax 运算（Tsai 等人，2019年），并将由此得到的注意力与稀疏变分高斯过程（SVGP）的均值联系起来。这一思路使我们能够应用 SVGP 方程进行不确定性估计；此外，我们还引入了解耦诱导点，以提高计算效率。

3.2.1 注意力作为稀疏变分高斯过程的均值

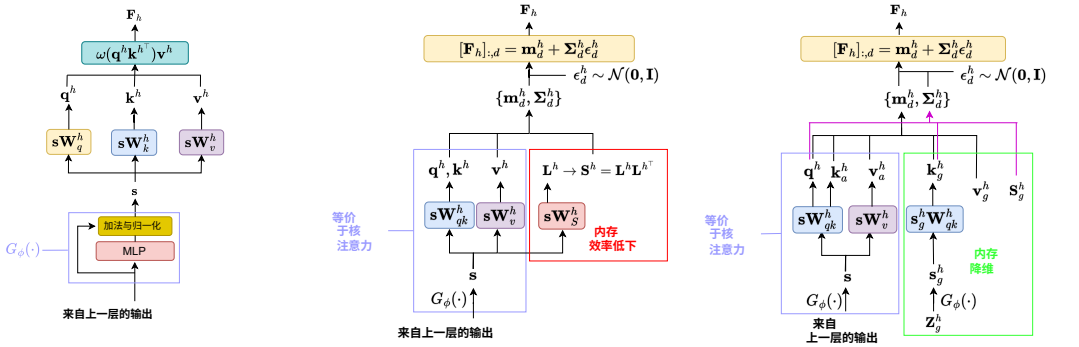
标准 Transformer 使用基于缩放点积的注意力块（Vaswani 等人，2017年）。给定查询 $\boldsymbol{q} \in \mathbb{R}^{T\times d_q}$、键 $\boldsymbol{k} \in \mathbb{R}^{T\times d_k}$和值 $\boldsymbol{v} \in \mathbb{R}^{T\times d_v}$，缩放点积（SDP）注意力定义如下：

$$\text{SDP-Attention:} \quad \boldsymbol{F} = \text{softmax}(\frac{\boldsymbol{q}\boldsymbol{k}^\top}{\sqrt{d_k}})\boldsymbol{v}, \tag{3.1}$$

其中，d_k $(= d_q)$ 是键的维度。由于注意力涉及测量 $\boldsymbol{q}$ 与 $\boldsymbol{k}$之间的相似度，Tsai 等人（2019年）通过将式 3.1 中的 $\text{softmax}(\frac{\boldsymbol{q}\boldsymbol{k}^\top}{\sqrt{d_k}})$ 替换为使用有效的对称核 $k(\cdot,\cdot)$ 计算得到的核 Gram 矩阵 $\boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}}$ $([\boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}}]_{i,\,j} = k(\boldsymbol{q}_i,\ \boldsymbol{k}_j))$，对 SDP 注意力进行了推广。我们将其称为核注意力，简称 K-注意力：

$$K\text{-Attention:} \quad \boldsymbol{F} = \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}}\boldsymbol{v}. \tag{3.2}$$

回顾一下，式 2.36 中 SVGP 的后验均值（已在第 2.3.2节讨论）为 $\boldsymbol{m} = \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}}\boldsymbol{K}_{\boldsymbol{Z}\boldsymbol{Z}}^{-1}\boldsymbol{m}_{\boldsymbol{u}}$，在输入 $\boldsymbol{X}$ 上进行评估时即是如此。现在，我们将 SVGP 的变分均值参数重新参数化为 $[\boldsymbol{v}]_{:,\,d} := \boldsymbol{K}_{\boldsymbol{Z}\boldsymbol{Z}}^{-1}\boldsymbol{m}_{\boldsymbol{u}}$，对每个维度（$d$）而言，



(a) 原始 Transformer　(b) 标准 SGPA（我们的方法）　(c) 解耦 SGPA（我们的方法）

图3.1：（a）原始 Transformer、（b）基于标准 SGPA 的 Transformer 以及（c）基于解耦 SGPA 的 Transformer 中，某一层内多头自注意力的一个头（h）的示意图。

$\boldsymbol{v}$，并将输入位置和诱导点位置分别定义为查询和键：$\boldsymbol{q} := \boldsymbol{X}$，$\boldsymbol{k} := \boldsymbol{Z}$。这样，我们便可以将 SVGP 的后验均值改写为

$$\text{SVGP-mean} \quad \boldsymbol{m} = \boldsymbol{K}_{\boldsymbol{qk}}[\boldsymbol{v}]_{:,d} \tag{3.3}$$

由此，可以确定稀疏变分高斯过程的后验均值与核注意力模块输出的每个维度之间的等价关系。这使我们能够在后续章节中自然地扩展高斯过程及其可扩展近似的工具箱，以量化 Transformer 中的不确定性。

3.2.2 标准 SGPA 及其对自注意力的低效性

观察到 K-注意力与稀疏变分高斯过程均值之间的等价关系后，一个自然的不确定性估计思路是应用稀疏变分高斯过程技术来计算近似后验方差。具体而言，我们引入一组变分协方差参数$\boldsymbol{S} \in \mathbb{R}^{T\times T\times d_v}$（以 T作为键/诱导输入的数量），并使用证据下界（式2.38）对其进行优化，如第2.3.2 节所述（见附录B.1.1 的完整推导）。该过程返回后验注意力输出每个维度（d）的均值和协方差：

$$\boldsymbol{m}_d = \boldsymbol{K}_{\boldsymbol{qk}}[\boldsymbol{v}]_{:,d}, \quad \boldsymbol{\Sigma}_d = \boldsymbol{K}_{\boldsymbol{qq}} + \boldsymbol{K}_{\boldsymbol{qk}}(\boldsymbol{K}_{\boldsymbol{kk}}^{-1}[\boldsymbol{S}]_{:,:,d}\boldsymbol{K}_{\boldsymbol{kk}}^{-1} - \boldsymbol{K}_{\boldsymbol{kk}}^{-1})\boldsymbol{K}_{\boldsymbol{kq}}. \tag{3.4}$$

通过这种方式，我们针对注意力输出的每个维度独立拟合一个稀疏变分高斯过程：对于每个维度 d，使用相同的核、但采用不同的变分均值（$[\boldsymbol{v}]_{:,\,d}$）和协方差（$[S]_{:,\,:,\,d}$）参数，来拟合式3.4 中给出的稀疏变分高斯过程。我们将这种方法称为标准 *SGPA*，并在图3.1b中展示其运算过程。

遗憾的是，将标准 SGPA 应用于基于多头自注意力的 Transformer 时，计算效率会很低。在每个注意力层中，每个头的键 h, $\boldsymbol{k}^h$ 都是将前一层的输出输入神经网络后得到的。此外，查询和键所用的投影矩阵（定义见第 2.1.1 节）按照 Tsai 等人（2019）的方法共享（即 $\boldsymbol{W}_q^h = \boldsymbol{W}_k^h := \boldsymbol{W}_{qk}^h \in \mathbb{R}^{d_s \times d_k}$），以获得有效的对称核。因此，自注意力层中的查询和键相同；更重要的是，它们依赖于输入，即 $\boldsymbol{k}^h = \boldsymbol{s}\boldsymbol{W}_k^h$（其中 $\boldsymbol{s} \in \mathbb{R}^{T \times d_s}$ 是多头自注意力模块的输入），并会随着 Transformer 的输入序列变化而变化。因此，要将标准稀疏变分高斯过程框架（式 3.4）扩展到自注意力，协方差参数 $\boldsymbol{S}^h$ 也需要依赖于输入，以适应不断变化的诱导输入 $\boldsymbol{k}^h$。一种直接的想法是通过线性投影来参数化$\boldsymbol{S}^h$，例如，对于一个头，使用 $vec(\boldsymbol{L}^h) = \boldsymbol{s}\boldsymbol{W}_s^h$，其中 $\boldsymbol{L}^h$ 是 $\boldsymbol{S}^h$ 的 Cholesky 因子（见图 3.1b）。即使在各输出维度间共享变分协方差，这种做法也会导致每个头的内存开销达到 $O(T^2)$；而由于 $\boldsymbol{K}_{\boldsymbol{k}^h\boldsymbol{k}^h}$ 是依赖于输入的，每个输入序列还会为每个头带来 $O(T^3)$ 的求逆运行时间开销。因此，标准 SGPA 在内存和运行时间方面的效率都很低，尤其是对于较长的输入序列。

注 3.2.1。 不同于变分参数是自由优化变量的情况，请注意，在自注意力设置中，后验均值的不同参数化通常会定义不同的后验预测分布。为说明这一点，假设 MHSA 模块中某个头的值 $\boldsymbol{v}^h$ 通过投影 $\boldsymbol{v}^h = \boldsymbol{s}^h\boldsymbol{W}_v^h (\boldsymbol{W}_v^h \in \mathbb{R}^{d_s \times d_v})$ 获得，且重新参数化的稀疏变分高斯过程后验均值给出 $\boldsymbol{m}^h = \boldsymbol{K}_{q^hk^h}\boldsymbol{v}^h = \boldsymbol{K}_{q^hk^h}\boldsymbol{s}^h\boldsymbol{W}_v^h$。现在，给定规范后验均值参数化以及通过 $\tilde{\boldsymbol{v}}^h = \boldsymbol{s}^h\tilde{\boldsymbol{W}}_v^h$ 获得的相应值，为了使所有 N 数据集中的序列都得到与重新参数化后的后验均值相同的结果，我们需要求解 $\tilde{\boldsymbol{W}}_v^h$，其中包含 N 个如下形式的线性方程组

$$\boldsymbol{K}_{q^hk^h}\boldsymbol{K}_{k^hk^h}^{-1}\boldsymbol{s}^h\tilde{\boldsymbol{W}}_v^h = \boldsymbol{K}_{q^hk^h}\boldsymbol{s}^h\boldsymbol{W}_v^h \iff \left(\boldsymbol{K}_{k^hk^h}^{-1}\boldsymbol{s}^h\right)\tilde{\boldsymbol{W}}_v^h = \boldsymbol{s}^h\boldsymbol{W}_v^h. \tag{3.5}$$

因此，我们总共需要求解一个包含 $N \times T \times d_v$ 个方程和$d_s \times d_v$ 个变量的线性方程组。一般来说，由于 $N \times T \gg d_s$，该方程组无解。

3.3 通过解耦 SGPA 提升时间与内存效率

我们提出通过将解耦稀疏高斯过程近似（Cheng 和 Boots，2017年；Salimbeni 等人，2018年）扩展到自注意力 SGPA，来解决上述效率低下的问题。

3.3.1 解耦 SVGP 的预备知识

解耦SVGP（Cheng 和 Boots，2017年； Salimbeni 等人， 2018年）可以解释为带有诱导点结构化变分分布的SVGP。假设我们将诱导点分成两组：$\{(\boldsymbol{z}_g^{(m)}, u_g^{(m)})\}_{m=1}^{M_g}$ 和$\{(\boldsymbol{z}_a^{(m)}, u_a^{(m)})\}_{m=1}^{M_a}$。原始解耦稀疏变分高斯过程（DSVGP；（Cheng 和 Boots，2017年））在 $\boldsymbol{u}$ 上考虑结构化变分分布：$= \boldsymbol{u}_{g\cup a} = \boldsymbol{u}_g \cup \boldsymbol{u}_a$:

$$q(\boldsymbol{u}) = q(\boldsymbol{u_g})q(\boldsymbol{u}_a|\boldsymbol{u}_g) = \mathcal{N}(\boldsymbol{u}_g; \mathbf{0}, \boldsymbol{S}_g)\mathcal{N}(\boldsymbol{u}_a; \boldsymbol{C}\boldsymbol{u}_g + \boldsymbol{m}_a, , \boldsymbol{B}), \tag{3.6}$$

其中 $\boldsymbol{B} = \boldsymbol{K}_{Z_aZ_a} - \boldsymbol{K}_{Z_aZ_g}\boldsymbol{K}_{Z_gZ_g}^{-1}\boldsymbol{K}_{Z_gZ_a}$ 和 $\boldsymbol{C} = \boldsymbol{K}_{Z_aZ_g}\boldsymbol{K}_{Z_gZ_g}^{-1}$。相应的变分均值和协方差为:

$$\boldsymbol{m_u} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{m}_a \end{pmatrix}, \quad \boldsymbol{S_u} = \begin{pmatrix} \boldsymbol{S}_g & \boldsymbol{S}_g\boldsymbol{C}^\top \\ \boldsymbol{C}\boldsymbol{S}_g & \boldsymbol{B} + \boldsymbol{C}\boldsymbol{S}_g\boldsymbol{C}^\top \end{pmatrix}. \tag{3.7}$$

这样，稀疏变分高斯过程的后验预测协方差中所使用的某些项便可被消去:

$$\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{S_u} = \begin{bmatrix} (\boldsymbol{K}_{Z_gZ_g}^{-1}\boldsymbol{S}_g - \boldsymbol{I})\boldsymbol{K}_{Z_gZ_g}^{-1}\boldsymbol{K}_{Z_aZ_g} & \boldsymbol{K}_{Z_aZ_a}^{-1}\boldsymbol{S}_g \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix}, \tag{3.8}$$

$$\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{S_u}\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1} - \boldsymbol{K}_{\boldsymbol{ZZ}}^{-1} = \begin{bmatrix} \boldsymbol{K}_{Z_gZ_g}^{-1}\boldsymbol{S}_g\boldsymbol{K}_{Z_gZ_g}^{-1} - \boldsymbol{K}_{Z_gZ_g}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{3.9}$$

将这种结构化的变分均值和协方差参数代入稀疏变分高斯过程的后验预测分布（式 2.36），即可得到后验分布 $q(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{Z})$，用于 DSVGP，它是一个高斯分布，具有如下均值$\boldsymbol{m_f}$ 和协方差 $\boldsymbol{\Sigma_{ff}}$:

$$\begin{aligned} \boldsymbol{m_f} &= \boldsymbol{K}_{\boldsymbol{X}Z_a}\boldsymbol{K}_{Z_aZ_a}^{-1}\boldsymbol{m}_a, \\ \boldsymbol{\Sigma_{ff}} &= K_{\boldsymbol{XX}} + \boldsymbol{K}_{\boldsymbol{X}Z_g}\boldsymbol{K}_{Z_gZ_g}^{-1}(\boldsymbol{S}_g - \boldsymbol{K}_{Z_gZ_g})\boldsymbol{K}_{Z_gZ_g}^{-1}\boldsymbol{K}_{Z_g\boldsymbol{X}}. \end{aligned} \tag{3.10}$$

显然，用于计算该后验预测均值和协方差的诱导点现在已完全解耦。此外，由于在后验预测分布$q(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{Z})$中，变分均值参数 $\boldsymbol{m}_a$ 总是与 $\boldsymbol{K}_{Z_aZ_a}^{-1}$的左侧矩阵乘法一起出现，因此我们可以将变分均值重新参数化为$\boldsymbol{v}_a := \boldsymbol{K}_{Z_aZ_a}^{-1}\boldsymbol{m}_a$，并将其作为最终的变分参数进行优化。这样，后验预测均值便可改写为 $\boldsymbol{m_f} = \boldsymbol{K}_{\boldsymbol{X}Z_a}\boldsymbol{v}_a$。

Ort正交解耦 SVGP（ODSVGP；（Salimbeni 等人，2018年））考虑了另一 er

s两组诱导点的结构化高斯分布变分分布 :

$$\begin{aligned} q(\boldsymbol{u}) &= q(\boldsymbol{u_g})q(\boldsymbol{u}_a|\boldsymbol{u}_g) \\ &= \mathcal{N}(\boldsymbol{u}_g; \boldsymbol{m}_g, \boldsymbol{S}_g)\mathcal{N}(\boldsymbol{u}_a; \boldsymbol{C}\boldsymbol{u}_g + \boldsymbol{B}\boldsymbol{K}_{\boldsymbol{Z_a}\boldsymbol{Z_a}}^{-1}\boldsymbol{m}_a, \boldsymbol{B}), \end{aligned} \tag{3.11}$$

相应的变分均值和协方差为:

$$\boldsymbol{m_u} = \begin{pmatrix} \boldsymbol{m}_g \\ \boldsymbol{C}\boldsymbol{m}_g + \boldsymbol{B}\boldsymbol{K}_{\boldsymbol{Z}_a\boldsymbol{Z}_a}^{-1}\boldsymbol{m}_a \end{pmatrix}, \quad \boldsymbol{S_u} = \begin{pmatrix} \boldsymbol{S}_g & \boldsymbol{S}_g\boldsymbol{C}^\top \\ \boldsymbol{C}\boldsymbol{S}_g & \boldsymbol{B} + \boldsymbol{C}\boldsymbol{S}_g\boldsymbol{C}^\top \end{pmatrix}. \tag{3.12}$$

再次将上述 $\boldsymbol{m_u}$ 和 $\boldsymbol{S_u}$ 代入式 2.36，消去部分项后，我们可以得到 ODSVGP 对 $\boldsymbol{f}$ 的后验预测分布（式 3.8 和 3.9 ）。这是一个高斯分布，其均值为 $\boldsymbol{m_f}$，协方差为 $\boldsymbol{\Sigma_{ff}}$：

$$\begin{aligned} \boldsymbol{m_f} &= \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}_a}\boldsymbol{K}_{\boldsymbol{Z}_a\boldsymbol{Z}_a}^{-1}\boldsymbol{m}_a - \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}_g}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g}^{-1}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_a}\boldsymbol{K}_{\boldsymbol{Z}_a\boldsymbol{Z}_a}^{-1}\boldsymbol{m}_a + \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}_g}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g}^{-1}\boldsymbol{m}_g, \\ \boldsymbol{\Sigma_{ff}} &= K_{\boldsymbol{X}\boldsymbol{X}} + \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}_g}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g}^{-1}(\boldsymbol{S}_g - \boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g})\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g}^{-1}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{X}}. \end{aligned} \tag{3.13}$$

再次对变分均值参数进行重新参数化，$\boldsymbol{v}_a := \boldsymbol{K}_{\boldsymbol{Z}_a\boldsymbol{Z}_a}^{-1}\boldsymbol{m}_a$ 和$\boldsymbol{v}_g := \boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g}^{-1}\boldsymbol{m}_g$，即可得到 ODSVGP 的后验预测均值和协方差的最终表达式:

$$\begin{aligned} \boldsymbol{m_f} &= \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}_a}\boldsymbol{v}_a - \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}_g}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g}^{-1}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_a}\boldsymbol{v}_a + \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}_g}\boldsymbol{v}_g, \\ \boldsymbol{\Sigma_{ff}} &= K_{\boldsymbol{X}\boldsymbol{X}} + \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{Z}_g}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g}^{-1}(\boldsymbol{S}_g - \boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g})\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{Z}_g}^{-1}\boldsymbol{K}_{\boldsymbol{Z}_g\boldsymbol{X}}. \end{aligned} \tag{3.14}$$

请注意，与 DSVGP 不同，ODSVGP 的后验预测均值是基于两组诱导点计算的。在超参数调优阶段进行的初步实验中，我们发现，将 ODSVGP 应用于 SGPA 框架时，其效果优于 DSVGP（见附录 B.3.1），因此我们重点研究将 ODSVGP 集成到 SGPA 中，并从现在起将由此得到的方法称为解耦 SGPA。

3.3.2 解耦 SGPA

在 SGPA 框架中应用 ODSVGP 时，除了依赖于输入的（或“摊销的”）键/诱导输入 $\boldsymbol{k}^h$（此后我们称之为 $\boldsymbol{k}_a^h$）之外，对于每个头 h，我们还另外纳入 M_g 个“全局”键/诱导输入 $\boldsymbol{k}_g^h$，它们在所有输入序列之间共享。主要思想是仅使用全局键来计算稀疏高斯过程的方差，从而使$\boldsymbol{S}^h$ 矩阵的变分参数独立于输入序列。事实上，按照上一段给出的式3.14，我们可以计算每个输出维度（d）对应每个头的均值和协方差（我们省略上标 h 以便更加

表 3.1：标准 SGPA 与解耦 SGPA 的复杂度比较。

模型	Time	额外内存
MLE	$O(BT^2)$	-
标准 SGPA	$O(BT^3)$	$O(T^2)$
解耦 SGPA	$O(BT^2M_g + M_g^3)$	$O(M_g^2)$

简洁记法）：

$$\begin{aligned}\boldsymbol{m}_d &= \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}_a}[\boldsymbol{v}_a]_{:,d} - \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}_g}\boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{k}_g}^{-1}\boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{k}_a}[\boldsymbol{v}_a]_{:,d} + \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}_g}[\boldsymbol{v}_g]_{:,d},\\ \boldsymbol{\Sigma}_d &= \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}_g}\boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{k}_g}^{-1}([\boldsymbol{S}_g]_{:,:,d} - \boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{k}_g})\boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{k}_g}^{-1}\boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{q}},\end{aligned} \tag{3.15}$$

其中 $\boldsymbol{v}_g \in \mathbb{R}^{M_g\times d_v}$， $\boldsymbol{S}_g \in \mathbb{R}^{M_g\times M_g\times d_v}$ 是与全局键 $\boldsymbol{k}_g$相关的变分参数， $\boldsymbol{v}_a \in \mathbb{R}^{T\times d_v}$ 通过投影 $\boldsymbol{v}_a = \boldsymbol{s}\boldsymbol{W}_v$计算得到。我们将此方法称为 解耦 ***SPGA*** ，如图 3.1c所示。

与标准 SGPA（式 3.4）相比（解耦 SGPA 中的 $\boldsymbol{k}_a^h$ 与标准 SGPA 中的 $\boldsymbol{k}^h$ 相同），我们可以看到，解耦 SGPA 的后验均值也包含两个额外项，用于考虑全局诱导点的影响。但更重要的是，两种 SGPA 方法的后验方差仅在所使用的键/诱导输入上有所不同（输入依赖键 $\boldsymbol{k}^h$ 与全局键 $\boldsymbol{k}_g^h$），这带来了**解 SGPA 的关键优势。由于式 3.15 中的后验协方差仅涉及全局诱导点，变分协方差不再需要依赖于输入，因此可以自由地参数化（Cholesky 因子） $\boldsymbol{S}_g^h$。现在，协方差部分的参数数量为 $O(M_g^2)$ 量级（而标准 SPGA 中为 $O(T^2)$），矩阵求逆的计算只需一次性付出 $O(M_g^3)$ 的代价（而标准 SPGA 对每个输入序列都要付出 $O(T^3)$ 的代价）。请注意，我们可以自由选择全局诱导点的数量 M_g；在实践中，对于本文所研究的实验，我们发现 $M_g = O(\frac{T_{avg}}{H})$ 通常就足够了，其中 T_{avg} 是训练输入序列的平均长度。表 3.1 总结了 Transformer 一个头中 SGPA 的时间复杂度（批次大小为 B）和所需的额外内存（参数数量）。我们还列出最大似然估计（MLE）作为参考（注意，MLE 的内存复杂度不取决于输入序列长度 T）。由于时间和内存开销的节省十分显著，我们主要在实验中评估解耦 SGPA；在正文其余部分，我们将简称解耦 SGPA 为 SGPA。

3.4 基于解耦 SGPA 的 Transformer

到目前为止，我们介绍了用于量化多头自注意力模块中不确定性的 SGPA 方法。在 Transformer 模型中，多个层

注意力块会被使用，接下来我们将介绍基于解耦 SGPA 构建 Transformer 模型的方法。请注意，由于 SGPA 等价于稀疏高斯过程，下面介绍的 Transformer 模型可以看作是对深度高斯过程的一种稀疏近似（Damianou 和 Lawrence，2013年；Salimbeni 和 Deisenroth，2017年），其中每一层都具有深度核。

我们的 Transformer架构主要遵循 Vaswani 等人 (2017年提出的架构。第 l 个 SGPA 层的输入是前一个 SGPA 层的输出 $\boldsymbol{F}^{l-1} \in \mathbb{R}^{T \times d^{l-1}}$。我们首先通过非线性映射 $G_{\phi^l}: \mathbb{R}^{d^{l-1}} \to \mathbb{R}^{d^l}$处理输入，然后进行投影以获得查询、摊销键、全局键和值。具体来说，对于每个头 h:

$$\boldsymbol{q}^{l,h} = \boldsymbol{k}_a^{l,h} = G_{\phi^l}(\boldsymbol{F}^{l-1})\boldsymbol{W}_{qk}^{l,h}, \quad \boldsymbol{k}_g^{l,h} = G_{\phi^l}(\boldsymbol{Z}_g^{l,h})\boldsymbol{W}_{qk}^{l,h}, \quad \boldsymbol{v}_a^{l,h} = G_{\phi^l}(\boldsymbol{F}^{l-1})\boldsymbol{W}_v^{l,h}, \tag{3.16}$$

其中，$\boldsymbol{Z}^l$、$_g{}^h \in \mathbb{R}^{M_g \times d^{l-1}}$ 是第 l 层的全局诱导位置，与 $\boldsymbol{F}^{l-1}$ 定义在相同的空间中。接着，我们应用基核 $K_{base}(\cdot,\cdot)$ 来计算核矩阵。这等价于使用定义在 $\boldsymbol{F}^{l-1}$ 空间上的深度核，而 G_{ϕ^l} 的参数则视为该深度核的超参数。最后，利用与全局诱导位置 $\boldsymbol{Z}^l$、$_g{}^h$ 相关联的变分参数（$\boldsymbol{v}^l$、$_g{}^h$、$\boldsymbol{S}^l$、$_g{}^h$），我们可以根据公式 3.15 得到 $\boldsymbol{m}^l$、$_d{}^h$ 和 $\boldsymbol{\Sigma}^l$、$_d{}^h$。随后，我们按照 Salimbeni 和 Deisenroth（2017年）所述，使用重参数化技巧为每个头生成输出样本，将不确定性传播到下一层:

$$[\boldsymbol{F}_h^l]_{:,d} = \boldsymbol{m}_d^{l,h} + \boldsymbol{L}_{\boldsymbol{\Sigma}_d}^{l,h}\boldsymbol{\epsilon}_d^{l,h}, \quad \boldsymbol{\epsilon}_d^{l,h} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{3.17}$$

其中 $\boldsymbol{L}^l$, $_{\boldsymbol{\Sigma}_d}{}^h$是 $\boldsymbol{\Sigma}^l$的 *Cholesky* 因子, $_d{}^h$。该 SGPA 层的最终输出 $\boldsymbol{F}^l \in \mathbb{R}^{T \times d^l}$ 通过与标准Transformer相同的方式进行线性组合得到（参见公式 2.4）。

用于训练变分参数与核参数的证据下界目标函数，是遵循深度高斯过程与加性高斯过程（Duvenaud 等人，2011年）的方法推导得到的。其核心思想是：由于采用 SGPA 的 MHSA 中每个头都是一个（稀疏）高斯过程，最终输出 $\boldsymbol{F}^l$也可视为（稀疏）高斯过程的加权求和，而该求和本身仍是一个高斯过程（Duvenaud 等人，2011年）。这使得我们可以在最终组合之前对各个注意力头进行变分近似，而不是直接对 $\boldsymbol{F}^l$ 过程进行近似（Sun 等人，2021年）。假设近似后验 q 关于 $\{\boldsymbol{F}_h^l\}_{h=1}^H$在 h 上分解，则对应于输入序列 $\boldsymbol{F}^0 := \boldsymbol{X}$ 的证据下界为

（推导见附录B.1.2）：

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q(\boldsymbol{F}^L|\boldsymbol{F}^0,\{\boldsymbol{k}_g^{l,h}\}_{l=1,h=1}^{L,H})}[\log p(\boldsymbol{Y}|\boldsymbol{F}^L)] - \sum_{l=1}^{L}\sum_{h=1}^{H}\mathbb{E}_{q(\boldsymbol{F}^l|\boldsymbol{F}^0,\{\boldsymbol{k}_g^{j,h}\}_{j=1,h=1}^{l,H}))}[\mathrm{KL}(q(\boldsymbol{u}_{a\cup g}^{l,h}|\boldsymbol{k}_g^{l,h},\boldsymbol{F}^{l-1})||p(\boldsymbol{u}_{a\cup g}^{l,h}|\boldsymbol{k}_g^{l,h},\boldsymbol{F}^{l-1}))]. \tag{3.18}$$

在实践中，我们借助蒙特卡洛来估计$\mathcal{L}_{ELBO}$，所用的是在逐层传递时利用重参数化技巧迭代生成的函数值样本（式3.17）。

3.5 实验

我们在跨模态的预测任务上评估 SGPA，实验设置如下。

- 数据集：CIFAR10 和 CIFAR100（图像分类（Krizhevsky 等人，2009年），计算机视觉任务）；CoLA（语言可接受性预测（Warstadt等人，2019年），自然语言处理任务）和 IMDB（情感分析，（Maas，2011年），自然语言处理任务）。

- 网络架构：计算机视觉任务采用视觉Transformer（ViT（Dosovitskiy，2021年））。核注意力方面，自然语言处理任务采用指数核（Tsai 等人，2019年），计算机视觉任务采用 ARD-RBF 核（Rasmussen 和 Williams，2006年）。我们还评估了基于缩放点积（SDP）注意力的 Transformer。与 Tsai 等人（2019年）一样，我们发现，在所考察的大多数任务中，核注意力往往优于 SDP 注意力，因此正文中不列出 SDP 注意力的结果。这些结果见附录中的表格B.7。

- 基线方法：我们将所提出的方法与以下“单模型”方法进行比较：最大似然估计（MLE）、贝叶斯推断方法，包括平均场变分推断（MFVI，（Blundell 等人，2015年））、蒙特卡洛 Dropout（MCD，（Gal 和 Ghahramani，2016年））、Kronecker分解末层拉普拉斯近似（KFLLLA）（Kristiadi 等人，2020年），以及谱归一化神经高斯过程（SNGP）（Liu 等人，2020年）。对于使用验证集的任务，我们还考虑温度缩放（TS）（Guo 等人，2017年），并将验证集用作校准集。对于计算机视觉任务，我们还考虑集成方法：将 SGPA 集成（SGPAE）与深度集成（DE）进行比较（Lakshminarayanan 等人，2017年）。由于自然语言处理任务在不同运行中采用不同的训练集（验证集）—测试集划分，因此我们不考虑此类任务中的集成模型。

- 评估与指标：我们考虑三种评估设置：分布内性能、分布外（OOD）鲁棒性和分布外检测。测试集上的指标包括各任务的预测准确率指标，以及不确定性校准指标，例如负预测对数似然（NLL）、期望校准误差（ECE）和最大校准误差（MCE）（Naeini 等人，2015年；Guo 等人，2017年）。附录 B.2对这些不确定性校准指标作了简要介绍。我们报告±5 次独立运行所得各项指标的均值和两个标准误。对于分布外检测任务，我们采用 ROC 曲线和精确率-召回率曲线下面积（分别为 AUROC 和 AUPR），并报告每种方法在全部 6 项分布外检测任务中按 AUROC 和 AUPR 计算的平均排名。

为确保比较公平，在每项任务中，所有模型均使用相同的架构和优化设置进行训练。所有模型均从头开始训练，未进行预训练。实验细节见附录 B.4 ，实际运行时间的比较见附录 B.5。表格中的结果也列于附录B.7。

3.5.1 分布内校准

我们在图 3.2的第一、第二、第三和第四行中，分别报告了图像分类（CIFAR10 和 CIFAR100，未使用数据增强）、情感分析（IMDB）和语言可接受性（CoLA）任务的分布内测试数据评估结果。对于 CoLA 数据集，预测准确率采用马修斯相关系数（MCC）（Matthews，1975）衡量，而非准确率，与 Warstadt等人 (2019年)一致。

所考察的所有“单模型”校准方法通常都能改善校准效果，但情感分析任务是个例外：KFLLLA 的校准表现甚至不如 MLE（尽管 KFLLLA 在语言可接受性（CoLA）任务上的校准效果最佳，但其表现因任务而异，稳定性不足）。MFVI 虽然通常能实现最低的校准误差，但在所有实验中都严重欠拟合数据。这并不可取，因为改善校准效果不应以预测正确性显著下降为代价。作为反例，通过预测边缘类别概率可以实现 ECE 为零的完美校准，但这种预测在实践中毫无用处。对于 CIFAR100 上的图像分类，KFLLLA 的 ECE 与 SGPA 相比具有竞争力，但其 NLL 更差，且 MCE 在所有方法中最差。总体而言，与其他“单模型”基线方法相比，SGPA 的表现最佳：它在所有任务上都能持续实现更好的校准，同时

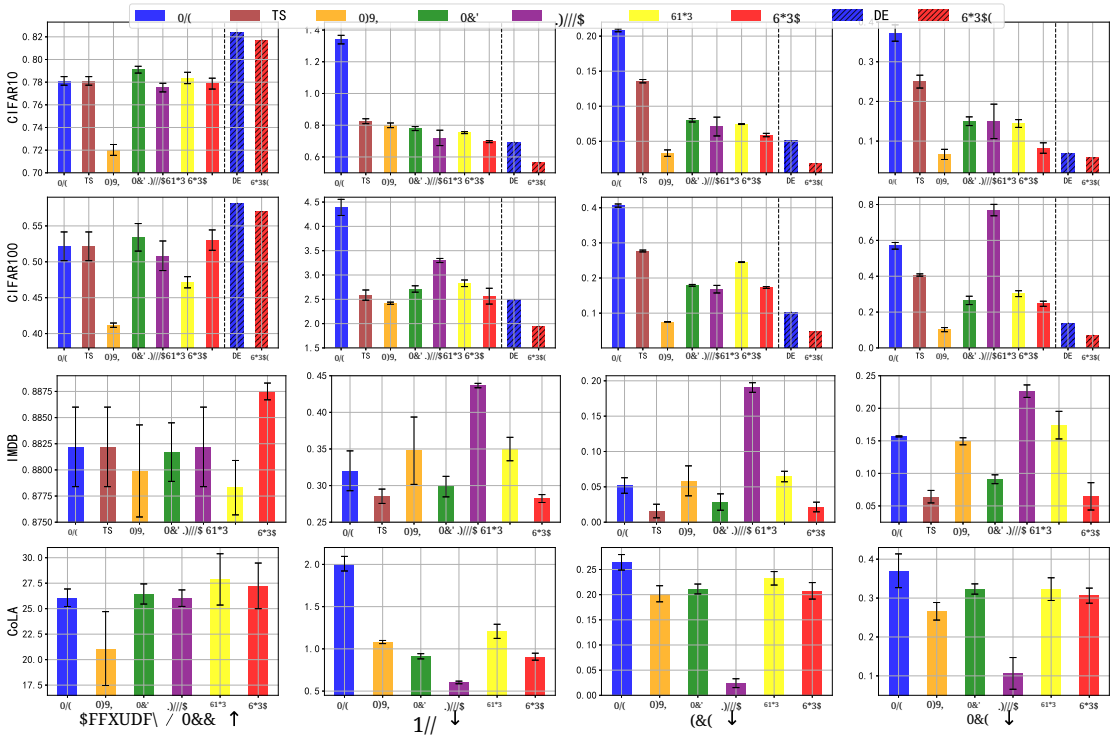


图 3.2：在 CIFAR10（第 1 行）、CIFAR100（第 2 行）、IMDB（第 3 行）和 CoLA（第 4 行）上训练的 Transformer 或 ViT 的测试集准确率（CoLA 使用马修斯相关系数）和校准指标。

保持有竞争力（甚至在 IMDB 上表现更佳）的预测准确率。与“单模型”方法相比，两种集成方法 DE 和 SGPAE 的预测准确率都高得多。SGPAE 在保持预测准确率具有竞争力的同时，校准表现明显优于 DE。

对于 CIFAR10 和 CIFAR100，我们还考虑了使用数据增强训练 ViT，并在图 3.3 中报告了分布内校准的结果。尽管某些“单模型”方法在部分情况下能够达到比深度集成和 SGPAE 更低的期望校准误差或最大校准误差，但在准确率和负对数似然方面，深度集成和 SGPAE 始终优于这些方法。

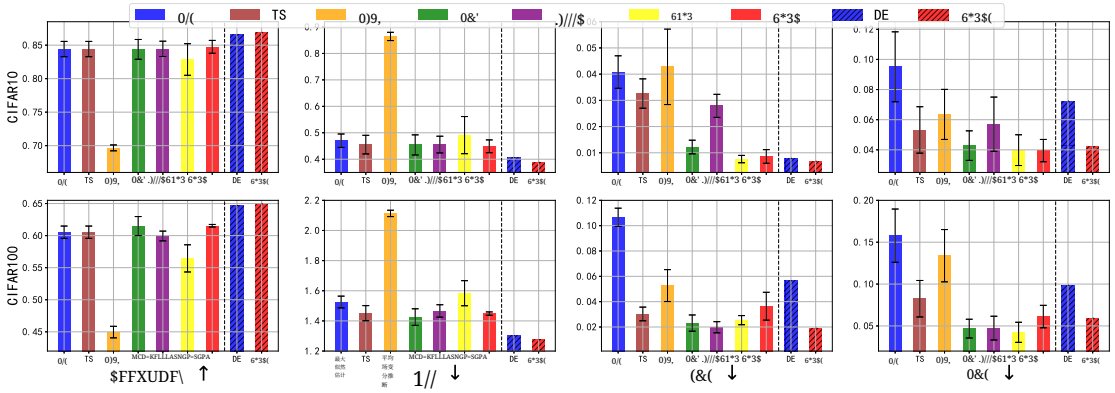


图 3.3：使用数据增强在 CIFAR10（上排）和 CIFAR100（下排）上训练的 ViT 的测试集准确率和校准指标。

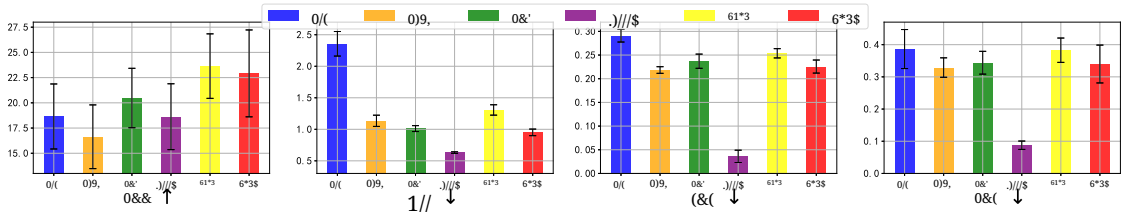


图 3.4：在使用 CoLA 训练的 Transformer 的分布外测试集上，测试集马修斯相关系数与校准指标。

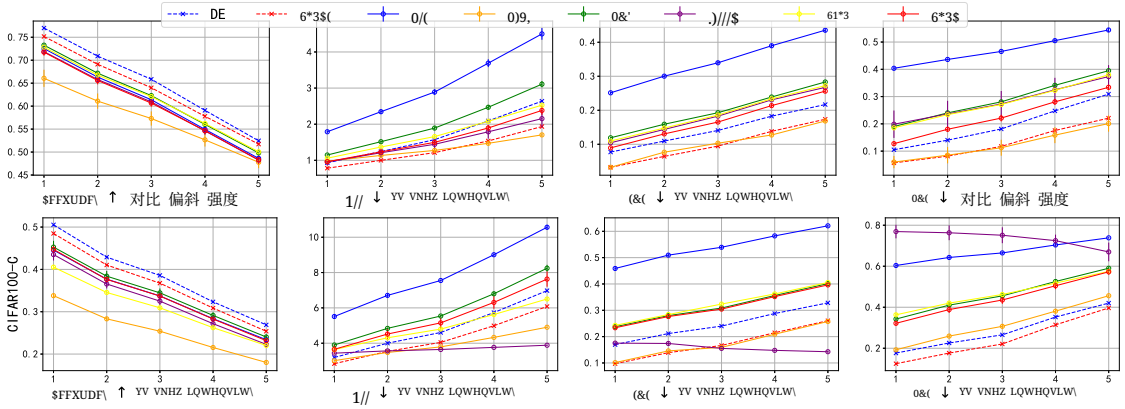


图 3.5：对于未使用数据增强、在相应干净数据上训练的 ViT，测试集准确率与校准指标随损坏偏斜强度的变化情况；上排为 CIFAR10-C，下排为 CIFAR100-C。

SGPAE 再次取得了总体最佳性能。在“单模型”方法中，平均场变分推断仍然对数据欠拟合；但对于其他方法，数据增强提升了模型性能，而 SNGP 的准确率相对较低。SGPA 与其他“单模型”基线方法之间的差异变小了，这可能是由于数据增强带来了较强的正则化效果。尽管如此，与时间序列、MCD 和 KFLLLA 相比，SGPA 通常返回更小的误差条，因此表现得更加稳健。

3.5.2 分布外数据上的鲁棒预测

接下来，我们评估 SGPA 在语言可接受性任务（CoLA）和图像分类任务（CIFAR10 与 CIFAR100）中面对分布偏移时的表现。CoLA 的分布外数据由 Warstadt等人 (2019年的同一批作者提供；对于 CIFAR 数据集，我们则使用受损 CIFAR 数据集（CIFAR10-C 和 CIFAR100-C）（Hendrycks 和 Dietterich, 2019年）作为分布外数据，其中包含含噪 CIFAR 图像。这些图像是在干净图像的基础上，以不同的偏斜强度引入了不同类型的失真。请注意，本节和下一节中，我们不考虑时间序列任务的分布外情形，因为 TS 是一种频率学派方法，仅用于校准分布内数据的不确定性。

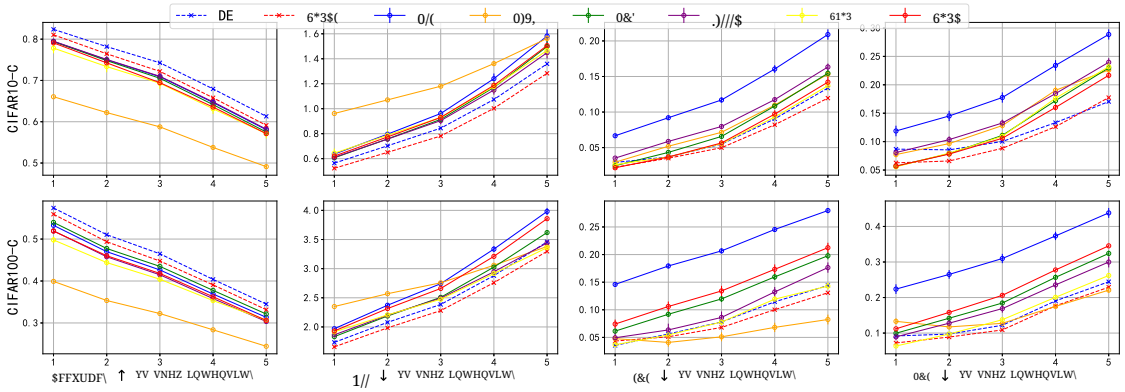


图3.6：使用数据增强在相应干净数据上训练的 ViT，在 CIFAR10-C（上排）和 CIFAR100-C（下排）上，测试集准确率与校准指标随损坏偏斜强度的变化。

图 3.4 报告了 CoLA 分布外测试的马修斯相关系数和校准指标。观测结果与分布内测试相似：在负对数似然和校准误差方面，SGPA 优于 MLE、MCD 和 SNGP，同时准确率也有所提高。MFVI 和 KFLLLA 的校准误差较低，但预测准确率不如 SGPA。尤其是，MFVI 再次出现了数据欠拟合。

针对图像分类的分布外鲁棒性测试，我们计算了受损 CIFAR 数据集上各项指标随偏斜强度的变化，并在图3.5中报告未使用数据增强时的结果，在图3.6中报告使用数据增强时的结果。未使用数据增强时，在“单模型”方法中，SGPA 的校准效果优于 MLE、MCD 和 SNGP，且不会损害准确率。MFVI 的校准误差低于 SGPA，但代价是出现欠拟合，尤其是在偏斜强度较小时。KFLLLA 的表现似乎不如 SGPA 稳定。对于 CIFAR10-C，SGPA 的校准效果优于 KFLLLA。对于 CIFAR100-C，KFLLA 的 NLL 和 ECE 最佳，但 MCE 最差。使用数据增强时，MFVI 仍然存在欠拟合，

表3.2：在 6 个分布外检测任务中，不同方法在 AUROC 和 AUPR 方面的平均排名

	未使用数据增强		使用数据增强	
模型	排名 (AUROC) ↓	排名 (AUPR) ↓	排名 (AUROC) ↓	排名 (AUPR) ↓
MLE	6.1667	5.5000	6.1667	6.0000
MFVI	7.0000	7.1667	8.0000	8.0000
MCD	5.6667	5.6667	4.1667	4.5000
KFLLLA	4.3333	4.5000	4.3333	4.0000
SNGP	5.3333	5.3333	6.3333	6.3333
SGPA	4.5000	4.6667	4.0000	4.1667
DE	1.6667	2.0000	1.8333	1.8333
SGPAE	**1.3333**	**1.1667**	**1.1667**	**1.1667**

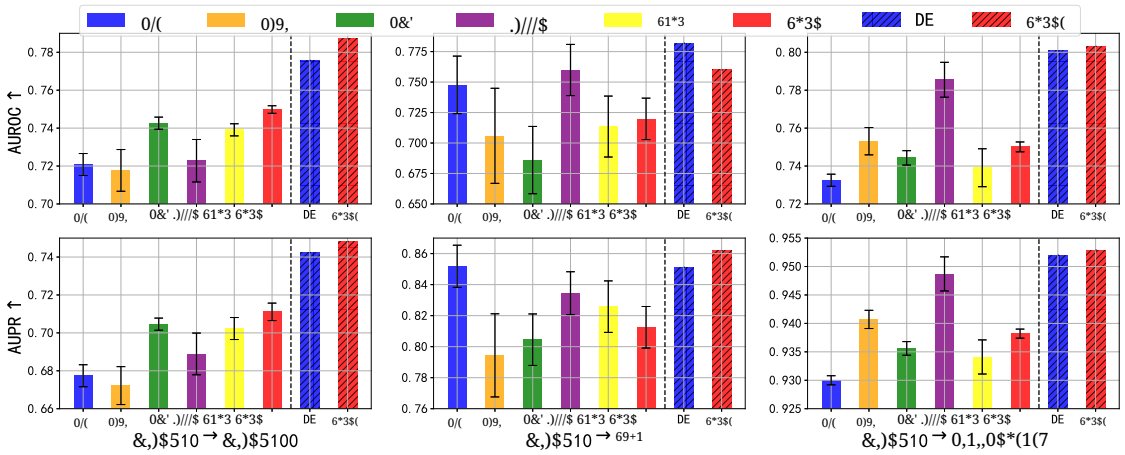


图3.7：使用未经过数据增强训练的 CIFAR10 视觉变换器进行分布外检测的 AUROC（上）和 AUPR（下）指标。

但 SGPA 与其他“单模型”基线方法之间的性能差距再次缩小。无论是否使用数据增强，集成方法的准确率都高于“单模型”方法；同时，SGPAE 的校准效果仍优于 DE，且准确率相近。

3.5.3 分布外检测

最后，我们考察在图像分类任务上训练的 Transformer 模型的分布外检测任务，以进一步评估不确定性估计的质量。在这里，我们使用预测熵为每个输入（来自分布内数据或分布外数据）评分，并根据熵是否低于或高于指定阈值，判定其属于“分布内”或“分布外”。在此检测任务中采用不同的阈值，使我们能够计算受试者操作特征（ROC）曲线和精确率-召回率（PR）曲线，并使用曲线下面积，即 AUROC 和 AUPR，作为性能

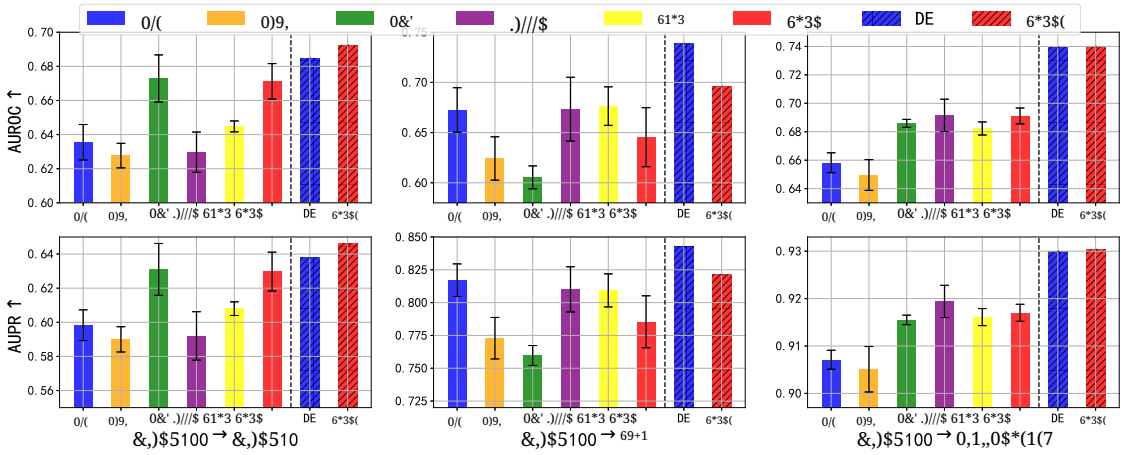


图3.8：使用未经过数据增强训练的 CIFAR100 视觉变换器进行分布外检测的 AUROC（上）和 AUPR（下）指标。

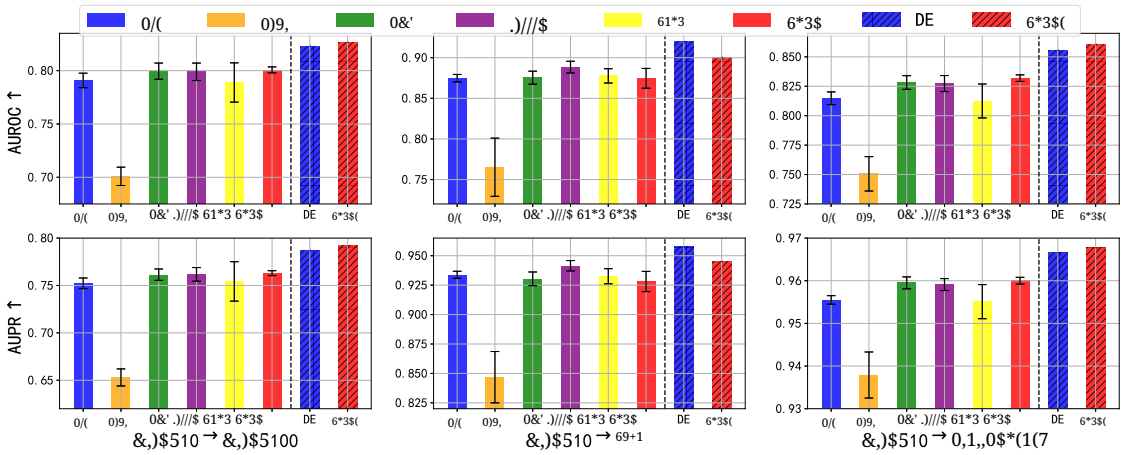


图3.9：使用在CIFAR10上经数据增强训练的视觉变换器进行分布外检测的AUROC（上）和AUPR（下）指标。

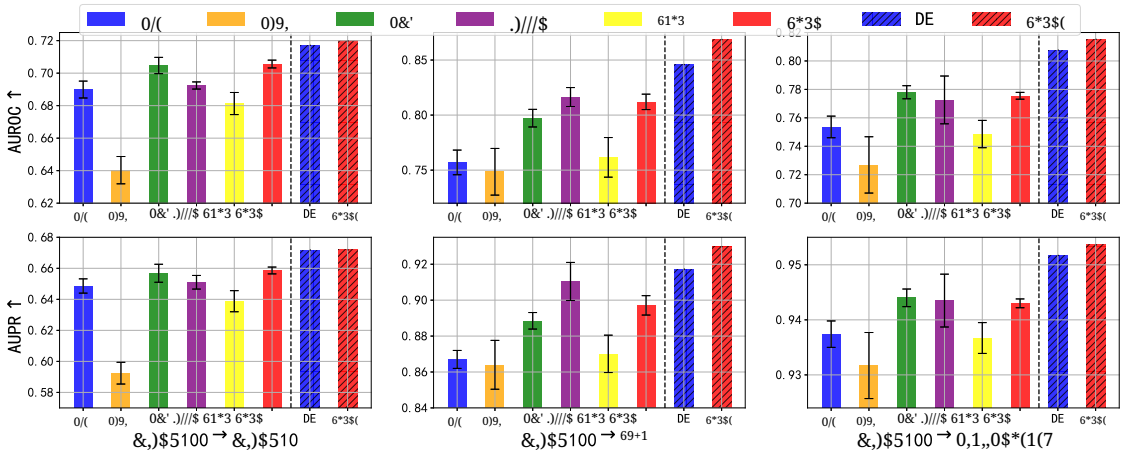


图3.10：使用在CIFAR100上经数据增强训练的视觉变换器进行分布外检测的AUROC（上）和AUPR（下）指标。

评估。

对于两个 CIFAR 数据集中的每一个，我们都将另一个 CIFAR 数据集、SVHN 和 mini-ImageNet 视为分布外数据集，因此总共构建了 6 个分布外检测任务。对于使用或不使用数据增强训练的每种方法，我们在表 3.2 中报告其在 6 个任务上的 AUROC 和 AUPR 平均排名。集成方法优于“单模型”方法，其中 SGPAE 的表现最佳。在“单模型”方法中，除表现最佳的 KFLLLA 外，SGPA 的表现优于所有其他基线方法。为了比较各任务中的方法，我们在图 3.7 和 3.8中绘制了每个任务中所有未使用数据增强训练的方法所达到的 AUROC 和 AUPR 值；在图 3.9和 3.10中，则绘制了每个任务中所有使用数据增强训练的方法所达到的 AUROC 和 AUPR 值。

3.5.4 附加结果总结

补充实验报告于附录 B.3 ，其中我们将主要结果总结如下。

- 在附录B.3.1中，我们发现，除了参数效率低下的问题之外，基于标准 SGPA 的 Transformer 也存在欠拟合。与解耦 SGPA 相比，标准 SGPA 在 CIFAR10 分类任务上的准确率明显更低。

- 在附录B.3.2中，我们报告了一项额外实验的结果，即使用 ZINC 数据集进行图属性回归（Dwivedi 等人，2020年）。对于这项任务，SGPAE 的表现最佳，而 SGPA 优于其他“单模型”基线方法。

3.6 相关工作

贝叶斯 Transformer。Tran 等（2019）和 Xue 等（2021）提出使用平均场变分推断，在 Transformer 的部分层的权重空间中进行近似后验推断。然而，在我们的实验中，我们发现这类方法会对数据欠拟合。Foong 等（2020）和 Coker 等（2022）从理论上证实了权重空间平均场变分推断存在这种欠拟合现象。另一类研究提出直接对注意力矩阵进行变分推断（Fan 等（2020）；Cinquin 等（2021））。然而，Fan 等（2020）只考虑了使用变分注意力进行微调，而 Cinquin 等（2021）只考虑了使用浅层网络在合成数据集或简单数据集上进行实验；此外，其拟合于注意力权重的变分分布在不同数据之间共享，这对于复杂问题而言可能限制过强。此外，他们发现，依赖于数据的注意力权重变分分布甚至会降低其方法的性能。Liu 等（2020）通过在最后一层的输出上拟合高斯过程（Bradshaw 等人，2017），考虑直接对 Transformer 输出进行贝叶斯推断。这种方法可以视为使用由 Transformer 定义深度核的高斯过程模型。与之不同的是，SGPA 拟合深度高斯过程，使不确定性能够通过 Transformer 的每个注意力层传播。此外，Liu 等（2020）提出为深度核保留距离感知性质。需要注意的是，这种距离保持技巧与我们的方法正交，也可以轻松集成到 SGPA 中。

相关高斯过程方法。 SGPA 的证据下界与 Sun 等人 (2021年的证据下界相似，他们也提出对加性高斯过程中的每个加性分量分别近似其后验分布（Duvenaud 等人，2011年）。区别在于核设计： Sun 等人 (2021年) 旨在将给定的核函数分解为正交的

“核基”，而在 SGPA 中，我们为每个注意力头考虑相同类型的核，但使用不同的核超参数。我们的方法也与稀疏中的稀疏高斯过程（SWSGP）有关（Tran 等人，2021年；Jafrasteh 等人，2022年），该方法允许针对每个数据点自适应地设置诱导点（类似于 SGPA 中依赖于输入的键 $\boldsymbol{k}_a$ ）。SGPA 与 SWSGP 之间的这一联系将在附录 B.6中进一步讨论。

3.7 结论与未来工作

我们提出了 SGPA，可直接对 Transformer 中注意力块的输出进行近似贝叶斯推断。与其他贝叶斯神经网络方法不同，SGPA 利用了注意力架构中蕴含的归纳偏置来构建贝叶斯 Transformer。此外，我们证明，与其他基线方法相比，基于 SGPA 的 Transformer 在预测准确率和校准之间实现了更好的平衡。我们还证明，SGPA 提升的不确定性估计质量有助于在分布偏移下保持鲁棒性，并进行分布外检测。

以下方向值得作为未来工作进行探索。首先，掩码预训练（Devlin 等人，2019年）已被证明对标准 Transformer 的下游任务至关重要，也可能提升基于 SGPA 的 Transformer 的性能。在本研究中，由于计算成本高昂，我们无法考虑预训练；此外，由于 SGPA 使用有效核替代了缩放点积，目前还没有可直接用于下游微调任务的预训练骨干网络。其次，许多现代 Transformer 任务要求以自回归方式进行下一个词元预测，这涉及基于解码器的架构（Vaswani 等人，2017年；Brown 等人，2020年；OpenAI 等人，2024b）。尽管本研究仅考虑基于编码器的 Transformer，但将 SGPA 扩展到基于解码器的 Transformer 将进一步扩大该方法的影响；我们在第6章提出了一种可能的解决方案。

第 4 章

在线域间高斯过程的循环记忆

本章基于Chen 等人（2025b）：

- Wenlong Chen*、Naoki Kiyohara*、Harrison Bo Hua Zhu*、Jacob Curran- Sebastian、Samir Bhatt 和 Yingzhen Li。**在线域间高斯过程的循环记忆**。载于神经处理系统进展（*NeurIPS*），2025。

主要思路由我提出，末位作者提供了有益的建议。三位共同第一作者（*）在末位作者的指导下，共同完成了代码实现、实验和论文撰写。此外，我完成了所有推导，并协助统筹其他作者的贡献。

正如我们在第 2.2.1 节中简要提到的，贝叶斯推断为在线学习或持续学习提供了自然的框架。本章中，我们再次通过利用深度序列模型中的归纳偏置，着重改进稀疏高斯过程模型中的近似在线贝叶斯推断。具体而言，我们提出 HiPPO-SVGP，它利用了 HiPPO（我们在第 2.1.2 节中介绍的一种强大的循环神经网络架构）的长程记忆保持能力，构建更优的稀疏近似，从而缓解在线稀疏高斯过程方法中的灾难性遗忘问题。

4.1 引言

高斯过程（GP）是时间序列建模的常用选择，因为它们具有较强的函数表达能力和不确定性量化能力（Roberts 等人，2013年；Fortuin 等人，2020年）。然而，高斯过程的计算和内存开销都很大

，计算复杂度分别为立方和二次。在在线回归场景中，例如天气建模，时间步数量可能非常庞大，使得高斯过程很快变得难以应用。尽管已有研究提出变分近似方法来解决计算复杂度问题，例如利用稀疏诱导点（稀疏高斯过程回归（Titsias，2009年）；稀疏变分高斯过程（Hensman 等人，2013年，2015年））和马尔可夫高斯过程（S¨arkk¨a 和 Solin，2019年；Wilkinson 等人，2021年），但每次有新数据到来时都从头重新拟合高斯过程模型，仍然会带来难以承受的开销。 Bui 等人（2017年）提出了一种在线稀疏变分高斯过程（OSVGP）学习方法，仅根据新到达的数据逐步更新高斯过程后验分布。然而，正如他们在论文中所指出的，其模型可能无法保留先前数据的信息，因为新数据到来时，诱导位置不可避免地会发生偏移。这是一个主要缺陷，因为除非不断增加诱导点的数量，否则其模型可能无法建模长期记忆。

在深度学习中，作为 Transformer 的替代方案（Vaswani 等人，2017年），研究者提出了许多关于状态空间模型（SSM）的重要工作，以对序列数据中的长期记忆进行建模。HiPPO（高阶多项式投影算子）框架最初旨在为循环神经网络注入长期记忆（Gu 等人，2020年），它为通过正交多项式投影将连续时间信号压缩到记忆状态中提供了数学基础。HiPPO 计算效率高，在长程记忆任务中表现出色，并构成了当前最先进的状态空间模型的基础，例如结构化状态空间序列（S4）模型（Gu 等人，2022年）和 Mamba（Gu 和 Dao，2023年；Dao 和 Gu，2024年）。

受 HiPPO 启发，我们提出在线 HiPPO 稀疏变分高斯过程（OHSVGP），将 HiPPO 框架应用于 SVGP，以利用其长程记忆建模能力。我们的方法将 HiPPO 时变正交投影视为域间 SVGP 的诱导变量（L´azaro-Gredilla 和 Figueiras-Vidal，2009年；Leibfried 等人，2020年；Van der Wilk 等人，2020年），其中基函数为时变正交多项式。我们表明，这种方法能够显著解决 OSVGP 中的记忆丢失问题，从而为将高斯过程应用于长期在线学习任务开辟了可能性。总而言之，我们的贡献包括：

- （第 4.3）我们通过将 HiPPO 投影解释为具有时变正交多项式基函数的诱导变量，证明 HiPPO 可融入域间高斯过程。这使诱导变量能够压缩历史数据，并捕获长期信息。

- （第 4.3.2 和 4.6.1 节）我们证明，核矩阵可以利用 th e

HiPPO框架的高效常微分方程演化，为OHSVGP带来额外一层计算效率。

- （第4.6节）我们在多种在线/持续学习任务上展示了 OHSVGP 的表现，包括时间序列预测、UCI基准上的持续学习，以及高斯过程变分自编码器中的持续学习。结果表明，在预测性能、长期记忆保持和计算效率方面，它优于其他在线稀疏高斯过程基线方法。

4.2 预备知识

本节简要介绍域间诱导点方法、使用稀疏高斯过程进行在线学习，以及高斯过程变分自编码器。

4.2.1 域间高斯过程

域间高斯过程与标准稀疏变分高斯过程几乎相同，唯一的区别是，前者并非将诱导变量设为在某些输入位置 $\boldsymbol{Z}$处评估的函数值，而是域间高斯过程（L´azaro-Gredilla 和 Figueiras-Vidal，2009年；Van der Wilk 等人，2020年）提出将诱导变量推广为

$$u_m := \int f(x)\phi_m(x)\mathrm{d}x, \tag{4.1}$$

其中 $\phi_m(x)$是基函数，以提供更大的灵活性。由于积分是线性算子，且 f 是高斯过程，因此 u_m 服从高斯分布[1]。函数值 $f_x := f(x)$与诱导变量之间的先验交叉协方差，以及诱导变量彼此之间的先验诱导协方差，都是通过积分而非直接进行核函数求值来计算的：

$$\begin{aligned}[\mathbf{K_{fu}}]_m &:= \mathrm{COV}(f_x, u_m) = \int k(x,x')\phi_m(x')\mathrm{d}x', \\ [\mathbf{K_{uu}}]_{nm} &:= \mathrm{COV}(u_n, u_m,) = \iint k(x,x')\phi_n(x)\phi_m(x')\mathrm{d}x\mathrm{d}x'.\end{aligned} \tag{4.2}$$

将标准先验协方差矩阵 $\boldsymbol{K_{XZ}}$ 和 $\boldsymbol{K_{ZZ}}$ 替换为 $\boldsymbol{K_{fu}}$和 $\boldsymbol{K_{uu}}$（如上所定义）后，域间高斯过程的后验预测分布和 ELBO 训练目标与第 2.3.2节所示的标准稀疏变分高斯过程完全相同。

我们可以看到，与基于诱导位置 $\mathbf{Z} \in \mathbb{R}^M$ 的标准稀疏变分高斯过程不同，后者可以根据训练数据移动位置，而域间高斯过程则绕过了

[1]事实上，域间 u 可以通过对先验高斯过程 $f(\cdot)$应用任意线性算子 L 来定义，条件是 $u := Lf(\cdot)$。在本研究中，我们重点关注 L 为积分算子的情况。

对诱导位置的选择，转而通过选择基函数来重新表述 ϕ_i。基函数决定了上述两个协方差/核矩阵的结构，进而调节高斯过程近似的函数空间。例如，一些域间高斯过程利用某些基函数的性质来稀疏化 $\boldsymbol{K}_{\boldsymbol{uu}}$，从而降低求逆的计算成本（Hensman 等人，2018年；Dutordoir 等人，2020年）。

4.2.2 在线稀疏高斯过程

考虑在线学习的情形，数据以批次的形式依次到达$\mathcal{D}_{t_1} := (\mathbf{X}_{t_1}, \mathbf{y}_{t_1}), \mathcal{D}_{t_2} := (\mathbf{X}_{t_2}, \mathbf{y}_{t_2}) \ldots$ 等等。例如，在时间序列预测场景中，数据会以间隔为 $(0, t_1), (t_1, t_2) \ldots$ 等等的形式到达。序贯贝叶斯推断为在线学习提供了一个有原则的框架，在该框架中，模型的旧后验分布可用作新任务的新先验。结合基于新数据的似然，我们即可得到新的后验分布。当真实的序贯后验更新难以处理时，同样需要进行近似。

本文关注使用稀疏高斯过程进行在线学习（第 2.3.2 节回顾了相关内容），即随着新数据到达，依次更新稀疏高斯过程后验分布。假设我们已经在第一批数据 D_{t_1} 上训练并获得了一个稀疏变分高斯过程（其诱导点为 $(\mathrm{Z}_{t_1}, \mathrm{u}_{t_1})$），$q_{t_1}(f) = \int p_{t_1}(f|\mathrm{u}_{t_1}) q_{t_1}(\mathrm{u}_{t_1}) d\mathrm{u}_{t_1}$，在线稀疏变分高斯过程（OSVGP；（Bui等人，2017））会在观察到第二个任务时，利用基于变分推断的近似来更新该稀疏变分高斯过程。基于更新后的诱导点集合 $(\mathrm{Z}_{t_2}, \mathrm{u}_{t_2})$ 得到的新稀疏变分高斯过程记为 $q_{t_2}(f) = \int p_{t_2}(f|\mathrm{u}_{t_2}) q_{t_2}(\mathrm{u}_{t_2}) d\mathrm{u}_{t_2}$。请注意，新稀疏变分高斯过程的先验也可以更新（即核超参数可以更新），因此 $p_{t_2}(f)$ 可能不同于 $p_{t_1}(f)$。最终，我们希望最小化全批次KL散度 $\mathrm{KL}\left(q_{t_2}(f) || p_{t_2}(f|\mathrm{D}_{t_1}, \mathrm{D}_{t_2})\right)$，这等价于最大化以下证据下界：

$$
\begin{aligned}
\mathcal{L}_{ELBO} &= \log \frac{p_{t_2}(\mathcal{D}_{t_1}, \mathcal{D}_{t_2})}{p_{t_1}(\mathcal{D}_{t_1})} - \mathrm{KL}\left(q_{t_2}(f) || p_{t_2}(f|\mathcal{D}_{t_1}, \mathcal{D}_{t_2})\right) \\
&= \int q_{t_2}(f) \log \frac{p_{t_2}(\mathbf{y}_{t_2}, \mathbf{y}_{t_1})}{p_{t_1}(\mathbf{y}_{t_1}) q_{t_2}(f)} p_{t_2}(f|\mathbf{y}_{t_1}, \mathbf{y}_{t_2}) df \\
&= \int q_{t_2}(f) \log \frac{p_{t_2}(f) p_{t_2}(\mathbf{y}_{t_2}|f) p_{t_2}(\mathbf{y}_{t_1}|f)}{p_{t_1}(\mathbf{y}_{t_1}) q_{t_2}(f)} df.
\end{aligned} \tag{4.3}
$$

注意到 $p_{t_1}(\mathcal{D}_{t_1})$ 在第一行第一个项的分母中是一个常数，因为它只依赖于旧的核超参数。不幸的是，在线设置下，由于不允许使用新模型重新访问旧数据，$p_{t_2}(\mathbf{y}_{t_1}|f)$ 是难以处理的。为绕过这一难题，Bui等人 (2017年) 提出将其

用以下内容近似这一项

$$p_{t_2}(\mathbf{y}_{t_1}|f) \approx p_{t_1}(\mathbf{y}_{t_1}|f) = \frac{p_{t_1}(f|\mathbf{y}_{t_1})p_{t_1}(\mathbf{y}_{t_1})}{p_{t_1}(f)} \approx \frac{q_{t_1}(f)p_{t_1}(\mathbf{y}_{t_1})}{p_{t_1}(f)}. \tag{4.4}$$

将这一近似代入式 4.3，得到“在线证据下界”：

$$\mathcal{L}^o_{ELBO} = \int q_{t_2}(f) \log \frac{p_{t_2}(f)p_{t_2}(\mathbf{y}_{t_2}|f)q_{t_1}(f)}{q_{t_2}(f)p_{t_1}(f)} df. \tag{4.5}$$

由于 $q_{t_1}(f)$ 和 $q_{t_2}(f)$ 分别完全由有限个诱导点（$\mathbf{u}_{t_1}$ 和 $\mathbf{u}_{t_2}$）确定，并且我们假设似然在各数据点之间可分解，因此目标函数可进一步简化为：

$$\begin{aligned} \mathcal{L}^o_{ELBO} = &\sum_{i=1}^{n_{t_2}} \mathbb{E}_{q_{t_2}(f_i)} \left[\log p_{t_2}(y_i \mid f_i)\right] + \mathrm{KL}\left(\tilde{q}_{t_2}(\mathbf{u}_{t_1}) \,\|\, p_{t_1}(\mathbf{u}_{t_1})\right) \\ &- \mathrm{KL}\left(\tilde{q}_{t_2}(\mathbf{u}_{t_1}) \,\|\, q_{t_1}(\mathbf{u}_{t_1})\right) - \mathrm{KL}\left(q_{t_2}(\mathbf{u}_{t_2}) \,\|\, p_{t_2}(\mathbf{u}_{t_2})\right), \end{aligned} \tag{4.6}$$

其中 $y_i \in \mathbf{y}_{t_2}$，对于 $i = 1, \ldots, n_{t_2}$，且 $\tilde{q}_{t_2}(\mathbf{u}_{t_1}) := \int p_{t_2}(\mathbf{u}_{t_1}|\mathbf{u}_{t_2})q_{t_2}(\mathbf{u}_{t_2})\mathbf{d}\mathbf{u}_{t_2}$。

遗憾的是，随着任务不断增多，OSVGP 可能无法捕获数据中的长期记忆，因为随着新数据的到来，无法保证优化后的诱导位置能够充分覆盖所有先前任务的输入域。

4.2.3 高斯过程变分自编码器

除了预测任务外，我们还研究高斯过程变分自编码器（GPVAE；（Casale 等人，2018年；Fortuin 等人，2020年；Ashman 等人，2020年；Jazbec 等人，2021年；Zhu 等人，2023年））中的持续学习。GPVAE 是一种基于高斯过程的生成模型，下面我们将简要介绍。GPVAE 在变分自编码器（VAE；（Kingma 和 Welling，2014 年））框架中嵌入了高斯过程。对于带有诱导变量的稀疏高斯过程，Jazbec 等人（2021年）提出了 SVGPVAE，将稀疏变分高斯过程（SVGP）与 VAE 形式相结合。似然 $p(y \mid \varphi_\theta(f))$ 由解码器网络 φ_θ 参数化，该网络以高斯过程潜变量样本 f 为输入，同时结合变分诱导后验 $q_\theta(\boldsymbol{u} \mid y)$。这个后验 $q_\theta(\boldsymbol{u} \mid \phi(y))$ 由编码器网络 ϕ 参数化。最后，潜在高斯过程 f 通常建模为具有独立分量的多输出高斯过程。研究表明，GPVAE 能够成功地对天气数据和视频等高维时间序列进行建模（Zhu 等人，2023年；Fortuin 等人，2020年）。在本研究中，我们在一组实验中采用 Jazbec 等人（2021年）定义的 SVGPVAE 模型；模型的详细规范和训练目标见附录 C.3。

4.3 基于 HiPPO 的域间诱导点高斯过程

我们将 HiPPO 框架与域间高斯过程联系起来，将由时变正交投影定义的 HiPPO 状态向量解释为域间诱导点。这样便能在保留长期记忆的同时，对高斯过程历史进行自适应压缩。

4.3.1 将 HiPPO 视为域间诱导变量

回顾第 4.2.1 节中的域间设定，诱导变量通过对一组基函数进行积分变换来定义。设 $f \sim$ 高斯过程$(0, k)$，并考虑时变基函数

$$\phi_m^{(t)}(x) = g_m^{(t)}(x)\omega^{(t)}(x), \tag{4.7}$$

其中，$g_m^{(t)}$是 HiPPO 的正交函数，$\omega^{(t)}$ 是相应的测度。我们将对应的域间诱导变量定义为

$$u_m^{(t)} = \int f(x)\phi_m^{(t)}(x)\mathrm{d}x, \tag{4.8}$$

它不像第 4.2.1 节中的变量那样是单维随机变量，而是随时间变化的随机函数（即随机过程）（$u_m^{(t)} := u_m(t)$），这是由时变基函数所致。这些诱导变量会随时间调整，通过 HiPPO 原理性的多项式投影，以紧凑形式捕获长程历史信息。

4.3.2 随时间调整核矩阵

在流式场景中，当较晚时刻有新的观测到来时，我们必须同时调整先验交叉协方差 $\boldsymbol{K_{fu}}$ 和诱导变量的先验协方差$\boldsymbol{K_{uu}}$。具体而言，我们的 HiPPO 构造中的基函数会随时间演变，因此相应的核量也需要更新。下面，我们将说明如何根据时间 t_1时的值，在新的时间 t_2 计算并更新这些矩阵。为清晰起见，我们先讨论 $\boldsymbol{K_{fu}}$，然后讨论 $\boldsymbol{K_{uu}}$。

先验交叉协方差 $\boldsymbol{K_{fu}^{(t)}}$。 回想一下，对于单个输入 x_n，其与第 m 个诱导变量的先验交叉协方差为

$$\left[\boldsymbol{K_{fu}^{(t)}}\right]_{nm} = \int k\left(x_n, x\right)\phi_m^{(t)}(x)\mathrm{d}x, \tag{4.9}$$

其形式与 HiPPO框架中的投影系数（式 2.7）相同。因此，我们可以利用相同的参数 $\boldsymbol{A}(t)$ 和 $\boldsymbol{B}(t)$，以符合 HiPPO 方法的方式计算 $\boldsymbol{K}_{\boldsymbol{fu}}^{(t)}$ 的时间演化。具体而言，

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{K}_{\boldsymbol{fu}}^{(t)}\right]_{n,:}=\boldsymbol{A}(t)\left[\boldsymbol{K}_{\boldsymbol{fu}}^{(t)}\right]_{n,:}+\boldsymbol{B}(t)k\left(x_n,t\right), \tag{4.10}$$

其中 $\left[\boldsymbol{K}_{\boldsymbol{fu}}^{(t)}\right]_{n,:}$ 是 n的第 $\boldsymbol{K}_{\boldsymbol{fu}}^{(t)}$ 行。矩阵 $\boldsymbol{A}(t)$ 和 $\boldsymbol{B}(t$取决于 HiPPO 测度和基函数的具体选择。在我们的实验中，我们采用 HiPPO-LegS，其显式矩阵形式见第 2.1.2节。随后，对 t 进行离散化（例如使用欧拉法或双线性变换），以获得递推更新规则。

先验诱导协方差 $\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}$。诱导变量的先验协方差矩阵的第 ℓm个元素为

$$\left[\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}\right]_{\ell m}=\iint k\left(x,x'\right)\phi_{\ell}^{(t)}(x)\phi_{m}^{(t)}(x')\mathrm{d}x\mathrm{d}x'. \tag{4.11}$$

由于 $k(x,\ x')$ 同时取决于 x 和 x'，基于原始 HiPPO 公式的递推更新规则无法直接用于 $\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}$。原始 HiPPO 公式是为单重积分设计的。幸运的是，对于平稳核，可以应用博赫纳定理（Rudin，1994）将二重积分分解为两个独立的单重积分，从而得到随机傅里叶特征（RFF）近似（Rahimi 和 Recht，2007年）：对于平稳核 $k(x,\ x')\ =k(|x-x'|)$，RFF 将其近似如下：

$$\begin{aligned}k(x,x')&=\mathbb{E}_{p(w)}\Big[\cos(wx)\ \cos(wx')+\sin(wx)\ \sin(wx')\Big]\\&\approx\frac{1}{N}\sum_{n=1}^{N}\left[\cos\left(w_nx\right)\cos\left(w_nx'\right)+\sin\left(w_nx\right)\sin\left(w_nx'\right)\right],\end{aligned} \tag{4.12}$$

其中 $w_n\sim p(w)$ 是核的谱密度[2]。将其代入双重积分（式 4.11）后，关于 x 和 x' 的依赖关系得以分离，从而将 $[\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}]_{\ell m}$化为一维积分乘积之和。每个基于蒙特卡洛样本 w 的积分具有以下形式之一

$$Z_{w,\ell}^{(t)}=\int\cos(wx)\phi_{\ell}^{(t)}(x)\,\mathrm{d}x\quad\text{or}\quad Z_{w,\ell}^{\prime(t)}=\int\sin(wx)\phi_{\ell}^{(t)}(x)\,\mathrm{d}x, \tag{4.13}$$

[2]对于非平稳核，可以采用更先进的傅里叶特征近似技术（例如（Ton 等人，2018年））。

这对应于 HiPPO框架中的标准投影系数（式 2.7）。我们进一步将基于 M 基函数的这些积分堆叠起来，并定义

$$\boldsymbol{Z}_w^{(t)} = \left[Z_{w,1}^{(t)}, \cdots, Z_{w,M}^{(t)}\right]^\top, \quad \boldsymbol{Z}_w'^{(t)} = \left[Z_{w,1}'^{(t)}, \cdots, Z_{w,M}'^{(t)}\right]^\top. \tag{4.14}$$

收集 N 个蒙特卡洛样本 $\{w_n\}_{n=1}^N$，我们构造特征矩阵

$$\boldsymbol{Z}^{(t)} = \left[\boldsymbol{Z}_{w_1}^{(t)} \quad \boldsymbol{Z}_{w_2}^{(t)} \quad \cdots \quad \boldsymbol{Z}_{w_N}^{(t)} \quad \boldsymbol{Z}_{w_1}'^{(t)} \quad \boldsymbol{Z}_{w_2}'^{(t)} \quad \cdots \quad \boldsymbol{Z}_{w_N}'^{(t)}\right], \tag{4.15}$$

协方差的 RFF近似为

$$\boldsymbol{K}_{\boldsymbol{uu}}^{(t)} \approx \frac{1}{N} \boldsymbol{Z}^{(t)} \left(\boldsymbol{Z}^{(t)}\right)^\top. \tag{4.16}$$

由于 $\boldsymbol{Z}_{w_n}^{(t)}$ 和 $\boldsymbol{Z}_{w_n}'^{(t)}$ 是标准的 HiPPO投影系数，其计算遵循与之前相同的 HiPPO常微分方程演化

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{Z}_{w_n}^{(t)} = \boldsymbol{A}(t)\,\boldsymbol{Z}_{w_n}^{(t)} + \boldsymbol{B}(t)\,h_n(t), \quad \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{Z}_{w_n}'^{(t)} = \boldsymbol{A}(t)\,\boldsymbol{Z}_{w_n}'^{(t)} + \boldsymbol{B}(t)\,h_n'(t), \tag{4.17}$$

其中 $h_n(t) = \cos(w_n t)$, $h_n'(t) = \sin(w_n t)$，这些常微分方程可以针对不同的蒙特卡洛样本并行求解。总之，该过程包括采样多个随机特征，将它们递归更新至时间 t，并对各样本取平均，以获得 $\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}$的 RFF近似。

或者，也可以直接对 $\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}$ 关于 t求导。这会得到一种不同于原始 HiPPO 表述形式的矩阵常微分方程。详情请参见附录 C.1。经验上，这种方法的朴素实现会出现数值不稳定（参见附录 C.1）。因此，我们的实验基于上述 RFF近似进行。

序贯变分更新。获得 $\boldsymbol{K}^{(t_2)}{}_{\boldsymbol{fu}}, \boldsymbol{K}^{(t_2)}{}_{\boldsymbol{uu}}$ 在新时间$t_2 > t_1$的结果后，我们按照第 4.2.2节所述的在线高斯过程框架执行变分更新。这确保了时间 t_2 的后验分布基于 $\boldsymbol{K}^{(t_1)}{}_{\boldsymbol{fu}}, \boldsymbol{K}^{(t_1)}{}_{\boldsymbol{uu}}$， 同时与新数据以及时间 t_1的先前后验分布保持一致。总体而言，该过程赋予基于HiPPO的域间高斯过程在线捕获长期记忆的能力。通过将诱导的核变换视为时间上的常微分方程，我们能够在以在线方式调整变分后验分布的同时，高效保留过去观测的记忆。图4.1b展示了在一个包含两个任务的玩具在线时间序列回归问题中，随着时间 x 增加，最优后验分布 $q(\boldsymbol{u}^{(x)})$的演化情况，其中 x 决定了先验交叉协方差矩阵和诱导协方差矩阵的循环更新结束时间（分别演化至 $\boldsymbol{K}_{\boldsymbol{fu}}^{(x)}$ 和 $\boldsymbol{K}_{\boldsymbol{uu}}^{(x)}$）。此外，当 $x > t_1$时，我们将在线更新 $q(\boldsymbol{u}^{(x)})$，并使用

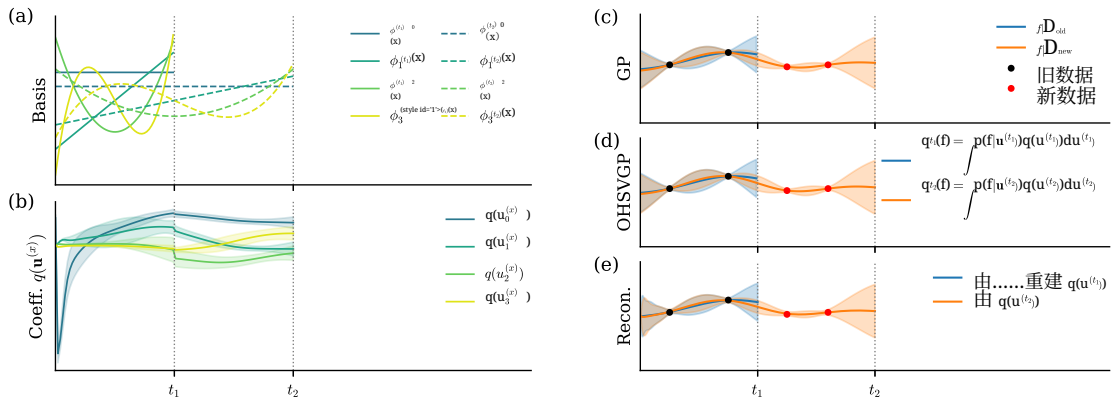


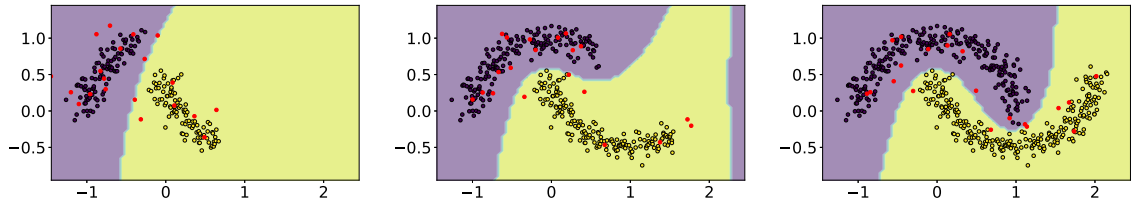
图 4.1：在包含 2 个任务的玩具时间序列上应用在线 HiPPO 稀疏变分高斯过程（OHSVGP）。其中，x 表示任意时间索引。(a) 终止时间索引为 $x = t_1$ 和 $x = t_2$时的时变基函数。(b)诱导变量最优近似后验分布的演变（均值 ± 2 边际标准差）。(c)、(d)和(e) 分别展示了后验在线高斯过程、OHSVGP 以及 OHSVGP 后验的有限基重构的预测均值 ± 2 标准差。

第二个任务中的两个数据点，通过优化在线证据下界（方程4.6）得到 $x = t_1$处的离散跳变。图4.1d展示了后验 OHSVGP，并将其与图4.1c中的金标准在线高斯过程拟合结果进行了比较。值得注意的是，如果$f \sim q_t(f)$，那么 $\int f(x)\phi_m^{(t)}(x)\mathrm{d}x \sim q(u_m^{(t)})$（详细推导见附录C.2）。因此，我们的框架还可顺带提供后验 OHSVGP 的有限基近似：$f = \sum_{m=1}^{M} u_m^{(t)} g_m^{(t)}(x)$，$u_m^{(t)} \sim q(u_m^{(t)})$。图4.1e绘制了有限基近似/重建结果；对于这个简单示例，它与后验 OHSVGP非常接近。

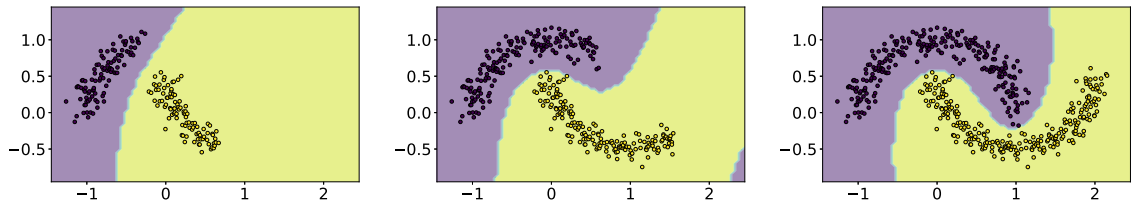
4.4 将 OHSVGP 扩展到多维输入

对于多维输入数据，假设第一批训练点的输入 $\{\boldsymbol{x}_n^{(1)}\}_{n=1}^{N_1}$按时间排序，使得 $\boldsymbol{x}_i^{(1)}$ 在 $\boldsymbol{x}_j^{(1)}$ 之后出现，当 $i > j$ 时；并进一步假设 $\boldsymbol{x}_i^{(1)}$ 出现在时间索引 $i\Delta t$ 处（即 $\boldsymbol{x}(i\Delta t) = \boldsymbol{x}_i^{(1)}$），其中 Δt 是用户指定的常数步长。在这种情况下，我们同样可以通过 HiPPO 递推得到域间先验协方差矩阵。例如，对式 4.10 中的常微分方程应用 $\boldsymbol{K}_{\boldsymbol{fu}}^{t}$ 前向欧拉法可得

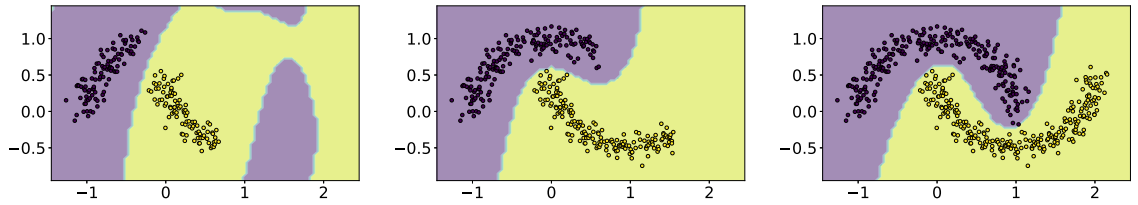
$$[\boldsymbol{K}_{\boldsymbol{fu}}^{((i+1)\Delta t)}]_{n,:} = [\boldsymbol{I} + \Delta t \boldsymbol{A}(i\Delta t)][\boldsymbol{K}_{\boldsymbol{fu}}^{(i\Delta t)}]_{n,:} + \Delta t \boldsymbol{B}(i\Delta t) k\left(\boldsymbol{x}_n^{(1)}, \boldsymbol{x}_i^{(1)}\right). \tag{4.18}$$



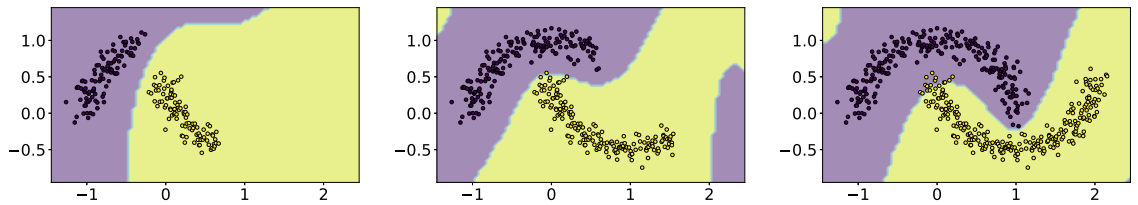
(a) OSVGP，完成任务 1 后　(b) OSVGP，完成任务 2 后　(c) OSVGP，完成任务 3 后



(d) OHSVGP-rand，之后任务 1　(e) OHSVGP-rand，之后任务 2　(f) OHSVGP-rand，任务 3



(g) OHSVGP-k-max，之后任务 1　(h) OHSVGP-k-max，之后任务 2　(i) OHSVGP-k-max，之后任务 3



(j) OHSVGP-k-min，之后任务 1　(k) OHSVGP-k-min，之后任务 2　(l) OHSVGP-k-min，之后任务 3

图 4.2：在双月数据集上，每个任务之后（共 3 个任务）OSVGP 和采用不同排序准则的 OHSVGP 模型的决策边界。对于 OSVGP，我们用红色标出诱导位置。

上面的方程可以看作是一个常微分方程的离散化（步长为 Δt），用于求解形式为 $\int_0^{N_1\Delta t} k\left(\boldsymbol{x}_n^{(1)}, \boldsymbol{x}(s)\right) \phi_m^{(t)}(s) ds$ 的路径积分。假设第 i 个训练输入 $\boldsymbol{x}_i^{(1)}$ 为 $\boldsymbol{x}_i^{(1)} := \boldsymbol{x}(i\Delta t)$，因此，基于与$\{\boldsymbol{x}(i\Delta t)\}_{i=1}^{N}$相对应的训练输入，通过离散递推近似求解路径积分。对于第二个任务，我们按照有序训练输入 $\{\boldsymbol{x}_n^{(2)N_2}{}_{n=1}$ 继续递推，将时间索引 $(N_1+i)\Delta t$ 分配给其第 i 个实例，并持续递推，直至连续学习完所有任务。在实践中，可以使用 Δt 的倍数作为步长，以加快递推；例如，不使用全部训练输入，而是仅基于$\{\boldsymbol{x}_1、\ \boldsymbol{x}_3、\ \boldsymbol{x}_5、\ \cdots\}$ 计算递推，步长为 $2\Delta t$。

当每个任务中的训练实例没有自然的时间顺序时（例如在标准的持续学习应用中），我们需要按照某种准则对实例进行排序，以创建伪时间顺序来适配 OHSVGP。这类似于将状态空间模型应用于非序列数据模态的做法；例如，状态空间模型应用于视觉任务时，会为图像中的图像块指定顺序，以便递推更新内存（Zhu 等人，2024年）。在我们的实验中，我们展示了 OHSVGP 应用于持续学习时的性能取决于所采用的排序准则。作为示例，我们通过一个二维持续分类问题，直观展示不同排序方法如何影响 OHSVGP 在持续学习中的性能。我们考虑使用 20 个诱导变量拟合 OHSVGP，以解决双月数据集上的持续二分类问题（Ganin 等人，2016年）。数据被划分为三个任务，我们使用伯努利似然对二元标签建模。我们考虑三种不同的排序准则：

- **随机，**记为 OHSVGP-rand。每个任务中数据点的顺序通过随机排列确定。

- **核相似度最大化，**记为 OHSVGP-k-max。我们选择任务 j 中的第 i 个点作为 $\boldsymbol{x}_{(j)\,i} = \arg\max_{\boldsymbol{x}\in\boldsymbol{x}_{(j)}} k(\boldsymbol{x}, \boldsymbol{x}_{(j)\,i-1})$，其中 $i>1$；并将第一个任务中的第一个点设为 $\boldsymbol{x}_{(1)\,1} = \arg\max_{\boldsymbol{x}\in\boldsymbol{x}_{(1)}} k(\boldsymbol{x}, \boldsymbol{0})$。其直觉是，在计算先验协方差矩阵时，当连续的 $\boldsymbol{x}$ 彼此接近，待记忆的信号往往会更加平滑。

- **核相似度最小化，**记为 OHSVGP-k-min。我们选择任务 j 中的第 i 个点作为 $\boldsymbol{x}_{(j)\,i} = \arg\min_{\boldsymbol{x}\in\boldsymbol{x}_{(j)}} k(\boldsymbol{x}, \boldsymbol{x}_{(j)\,i-1})$，其中 $i>1$；并将第一个任务中的第一个点设为 $\boldsymbol{x}_{(1)\,1} = \arg\min_{\boldsymbol{x}\in\boldsymbol{x}_{(1)}} k(\boldsymbol{x}, \boldsymbol{0})$。在这种情况下，我们刻意让先验协方差矩阵的循环计算难以记忆这些信号。

Figure 4.2 展示不同排序准则下 OHSVGP 在每个任务结束后的决策边界 s

我们还提供了 OSVGP 模型的决策边界作为参考。OHSVGP-k-max 和 OHSVGP-rand 均得到了准确率为 100% 的决策边界，而 OHSVGP-k-min 在任务 3 后出现了灾难性遗忘，这表明 OHSVGP 需要采用合理的排序准则，才能在持续学习任务中表现良好。

4.5 相关工作

在线稀疏高斯过程。以往研究主要关注通过不同的近似推断技术来降低稀疏近似误差，例如变分推断（Bui等人，2017年；Maddox 等人，2021年）、期望传播（Csat´o 和 Opper，2002年；Bui等人，2017年）、拉普拉斯近似（Maddox 等人，2021年），以及利用回放缓冲区增强的近似方法（Chang 等人，2023年）。在线更新诱导点这一相互独立的研究问题仍未得到充分探索，而主元 Cholesky 分解（Burt 等人，2019年）被Maddox 等人 (2021年)；Chang 等人 (2023年)所采用，是迄今在线更新诱导点最有效的方法之一。我们利用 HiPPO 的长期记忆能力来解决这一问题，设计了一种基于域间诱导变量的方法及其相应的基于递归的在线更新规则。值得注意的是，原则上，我们的 HiPPO诱导变量与上述所有近似推断框架兼容，因为与标准在线稀疏高斯过程相比，只有先验协方差矩阵的计算方式有所不同。

域间高斯过程。 据我们所知，OHSVGP 是在线学习领域中首个域间高斯过程方法。以往的域间高斯过程通常通过基于预定义测度的积分来构造诱导变量（例如，在固定区间上的均匀测度（Hensman 等人，2018年）或固定的高斯分布测度（L´azaro-Gredilla 和 Figueiras-Vidal，2009年）），以防止协方差发散；然而，这种预定义测度可能无法覆盖未来任务的时间索引所在的所有区域，因此不适用于在线学习。相比之下，OHSVGP 利用基于时变测度构造的自适应基函数，绕过了这一限制；随着更多任务到来，该测度会不断扩展至新的时间区域。

马尔可夫高斯过程。马尔可夫高斯过程（S¨arkk¨a and Solin，2019年；Wilkinson 等人，2021年）由于采用状态空间随机微分方程表示，在推断和训练过程中具有相似的递推结构。然而，马尔可夫高斯过程专为一维输入任务（如时间序列建模）设计，而我们将 OHSVGP 扩展到了多维输入任务。

4.6 实验

应用与数据集。我们在以下任务中将 OHSVGP 与基线方法进行评估。

- **时间序列预测**。我们考虑回归基准数据集：太阳辐照度（Lean，2004年）和音频信号（Bui 和 Turner，2014年），这些数据集取自 TIMIT 数据库（Garifolo 等人，1993）。我们按照Gal 和 Turner(2015年）和Bui等人(2017年）中描述的类似流程，分别对这两个数据集进行预处理（由于采用随机划分，训练集与测试集的划分有所不同）。此外，我们还考虑了巴西南部圣卡塔琳娜州的每日死亡人数时间序列，该数据涵盖了2020年3月至2021年2月的 COVID-19 疫情，数据取自Hawryluk等人(2021年）。我们将每个数据集依次划分为10个连续分区（COVID 数据集为5个），每个分区包含数量相同的训练实例，由此构建在线学习任务。

- **持续学习**。我们在两个具有多维输入的 UCI 数据集 Skillcraft（Blair 等人，2013年）和发电厂（Tfekci 和 Kaya，2014年）上进行持续学习，采用与 Stanton 等人(2021年）相同的数据预处理流程。我们首先根据数据点在第一维度上的值或其与原点之间的 L2 距离对数据点进行排序，再将排序后的数据集依次划分为10个连续任务，每个任务包含数量相同的训练实例，从而构建两类持续学习问题。

- **高维时间序列预测**。我们在 ERA5（Hersbach 等人，2023年）的逐小时气候数据上评估 GPVAEs。该数据集包含自2020年1月起英国各地随机分布的地点的17个变量，并被划分为10个连续任务，每个任务包含186个逐小时时间步。

基线。 我们将 OHSVGP 与 OSVGP（Bui等人，2017年）和 OVC（在线变分条件化；（Maddox 等人，2021年）进行比较。在每个任务开始时，OSVGP 通过从旧诱导位置和新数据输入中采样来初始化诱导位置，而 OVC 则通过主元 Cholesky 分解（Burt等人，2019年）对其进行初始化。我们同时考察两种做法：如 Chang 等人 （2023年）所示，固定已初始化的诱导位置（OVC）；或如 Maddox 等人 （2021年）所示，继续训练这些位置（OVC-optZ）。对于采用高斯似然的时间序列回归，我们考察 OHSGPR 和 OSGPR（分别为基于闭式 ELBO 的 OHSVGP 和 OSVGP），并进一步考察 OVFF（基于变分傅里叶特征（VFF）的 OSGPR，这是一种由 Hensman 等人（2018年）提出的跨域诱导点方法）。

超参数。 在每组实验中，所有模型均使用 Adam（Kingma 和 Ba，2015年）进行训练，且采用相同的学习率和迭代次数。对于 OHSVGP，我们基于 HiPPO-LegS（Gu 等人，2020年）构造诱导变量（使用其他 HiPPO 变体的可视化结果见附录 C.4.2 ），并使用 1000 个 RFF 样本。我们使用 ARD-RBF 核；但对于专为 Mat´ern 核设计的 OVFF，则改用 Mat´ern-$\frac{5}{2}$ 核。与 Maddox 等人（2021年）类似，我们发现在线持续更新核超参数并未带来性能提升。时间序列回归中采用可训练核超参数的结果列于附录 C.4.1 ，但当任务数量较多时，性能会变得不稳定。因此，我们要么只在初始任务期间训练核超参数，之后将其固定（第 4.6.3 节），要么从在初始任务上训练的完整高斯过程模型中获取这些超参数。还需注意，OVFF 需要将协方差计算为在一个预定义区间上的积分，该区间涵盖所有任务（包括未观测任务）的全部时间索引范围；这在实际在线学习场景中并不可行。在我们的实验中，该区间的两个端点分别设为所有任务数据点中的最小和最大时间索引。

评估与指标。 我们报告测试负对数似然（见附录 B.2），在高斯过程文献中通常称为负对数预测密度（NLPD）；对于 COVID 数据，则报告均方根误差（RMSE）和期望校准误差（ECE；（Naeini 等人，2015年；Guo 等人，2017年））。

$$\mathrm{NLPD} = -\frac{1}{N}\sum_{i=1}^{N}\log \hat{p}\big(y_i \mid x_i\big), \quad \mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(y_i - \hat{y}_i\big)^2}. \tag{4.19}$$

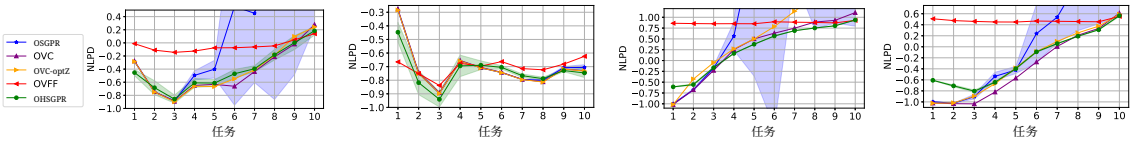
期望校准误差（（Naeini 等人，2015年；Guo 等人，2017年）；ECE）定义为置信度与覆盖率之间的偏差:

$$\mathrm{ECE} = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\Big(y_i \in \Big[\hat{q}_{\frac{1-c_k}{2}}(x_i)\ ,\ \hat{q}_{\frac{1+c_k}{2}}(x_i)\Big]\Big) \ -\ c_k,\right| \tag{4.20}$$

其中 $K = 10,$ $c_k \in \{0.05,\ 0.15,\ \ldots\ldots,\ 0.95\}$， N为测试点数量（x_i， y_i），而$\mathbf{1}(\cdot)$为指示函数。

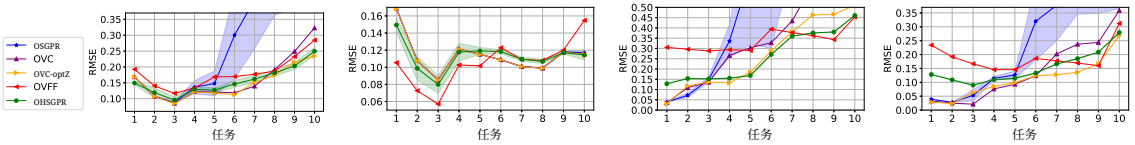
$\hat{}q_p(x_i)$ = 预测样本$\{\hat{}y_i^{(s)}\}_{s=1}^{S}$, $S = 100$.的经验 p–分位数

我们报告 5 次独立运行（ERA5 实验为 3 次）所得的均值及其对应的 95% 置信区间。



(a) 太阳能，M=50　(b) 太阳能，M=150　(c) 音频，M=100　(d) 音频，M=200

图4.3：太阳辐照度与音频信号预测数据集上，已学习任务的测试集负对数预测密度随已学习任务数量的变化。

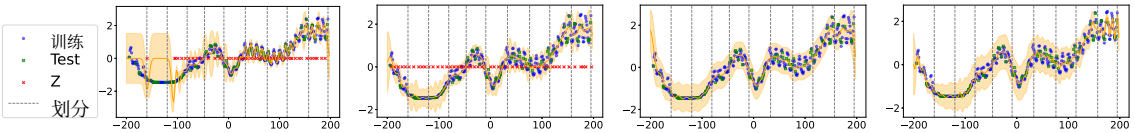


(a) 太阳能，M=50　(b) 太阳能，M=150　(c) 音频，M=100　(d) 音频，M=200

图 4.4：太阳辐照度和音频信号预测数据集上，已学习任务的测试集均方根误差随已学习任务数量的变化。

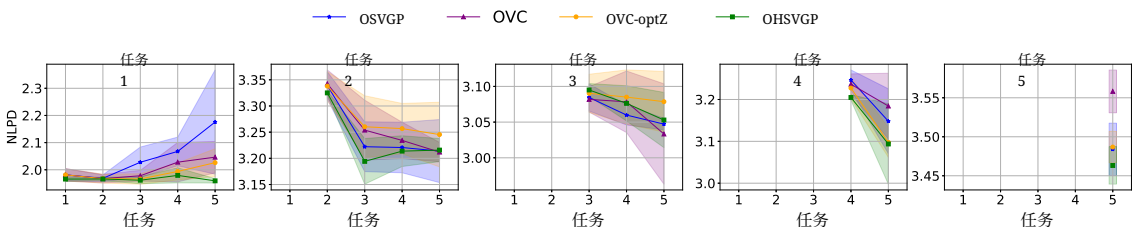
4.6.1 在线时间序列预测

时间序列回归。 图 4.3 和图4.4 分别展示了太阳辐照度和音频数据集上，不同方法在依次在线学习 10 个任务期间的负对数预测密度和均方根误差（针对过去的任务）。总体而言，OHSGPR 始终表现最佳，OVC 的表现也颇具竞争力；尤其随着学习的任务越来越多，这表明 OHSGPR 通过基于 HiPPO 的记忆机制有效保留了长期记忆。OSGPR 从任务 5 左右开始出现灾难性遗忘，尤其是在诱导点数量 M 较少时。尽管 OVC-optZ 与 OVC 一样，使用主元 Cholesky 分解初始化诱导位置，但经过进一步优化后，当 $M = 100$时，它在音频数据集上的表现从任务 6 开始下降。这表明在线证据下界目标无法保证诱导位置得到最优的在线更新以保留记忆。OVFF 往往在后期表现良好。然而，在最初的几个任务中，与其他方法相比，它对数据的拟合明显不足，因为它的诱导变量是通过在预定义区间上积分计算得出的，该区间涵盖了所有任务所在的区域；对于早期学习而言，这一区间过长，并非最优。

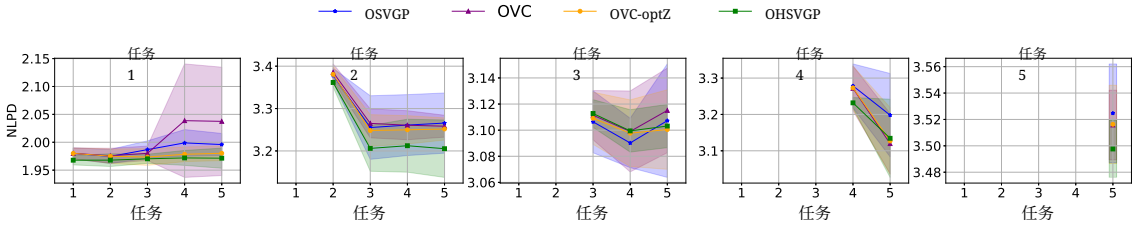


(a) OSGPR　(b) OVC　(c) OVFF　(d) OHSGPR

图 4.5：太阳能数据集任务 10 之后 OSGPR、OVC、OVFF 和 OHSGPR 的预测均值 ± 2 标准差。使用了 M $= 50$ 个诱导变量。



(a) M=15



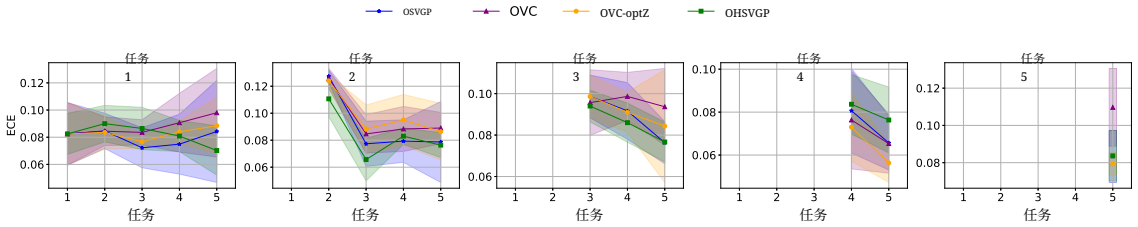
(b) M=30

图 4.6：在 COVID 数据集上对全部 5 个任务逐个进行持续学习后，每个任务的测试集负对数预测密度。

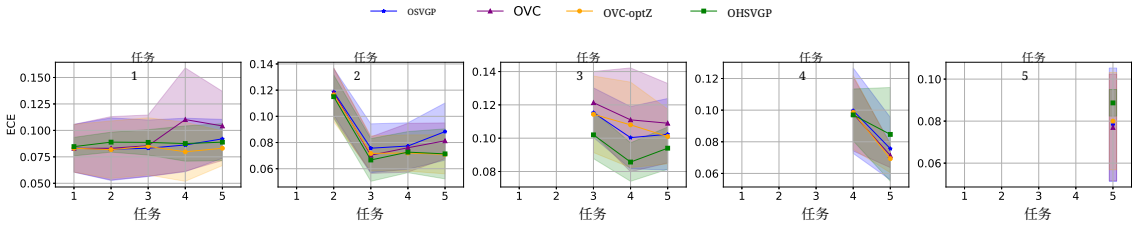
在图 4.5中，我们比较了太阳辐照度的全部 10 个任务完成在线学习后，不同方法得到的最终预测分布。OSGPR 的诱导位置 $\boldsymbol{Z}$ 在在线训练后往往会移动到后续任务所在的区域，而在缺乏足够诱导点的初始区域中，OSGPR 的预测会趋近于无信息先验高斯过程。相比之下，OHSGPR 在早期和近期时间段都保持了稳定的性能。

传染病建模 我们将高斯似然替换为负二项似然（非共轭），以捕捉 COVID-19 死亡人数中的过度离散。所有方法均使用 $M \in \{15,30\}$ 个诱导点，并针对每个任务训练 5000 次迭代，学习率为 0.01。图 4.6 和图 4.7 分别展示了全部五个任务在在线学习过程中 NLPD 和 ECE 的变化（即，在学习任务 i 之后，对任务 $j = i$、$i+1$、$\cdots$、5 的评估，适用于所有 i）。指标方差较大，反映了死亡人数数据的噪声特性，因为准确追踪 COVID-19 死亡人数十分困难。总体而言，OHSVGP 的表现最佳，而 OSVGP 在较小的 M。

运行时间比较。 表 4.1展示了不同方法学习全部任务所累积的实际运行时间。与必须迭代优化诱导位置的 OSVGP 和 OVC-optZ 不同（例如，针对时间序列回归任务，我们训练 1000 次迭代），OHSVGP、OVFF（二者均基于域间诱导点）以及 OVC（基于对每个任务的诱导位置进行一次性主元 Cholesky 更新）绕过了这一繁琐的优化。具体而言，OHSVGP 会递推演化 $\boldsymbol{K}_{\boldsymbol{fu}}$ 和 $\boldsymbol{K}_{\boldsymbol{uu}}$ 以应对每个新任务。对于回归问题，其中



(a) M=15



(b) M=30

图 4.7：在 COVID 数据集上对全部 5 个任务逐一进行持续学习后，各任务测试集的期望校准误差。

表 4.1：不同模型在时间序列预测实验中，于单块 NVIDIA RTX3090 GPU 上学习全部任务所需的墙钟累计运行时间（秒）。

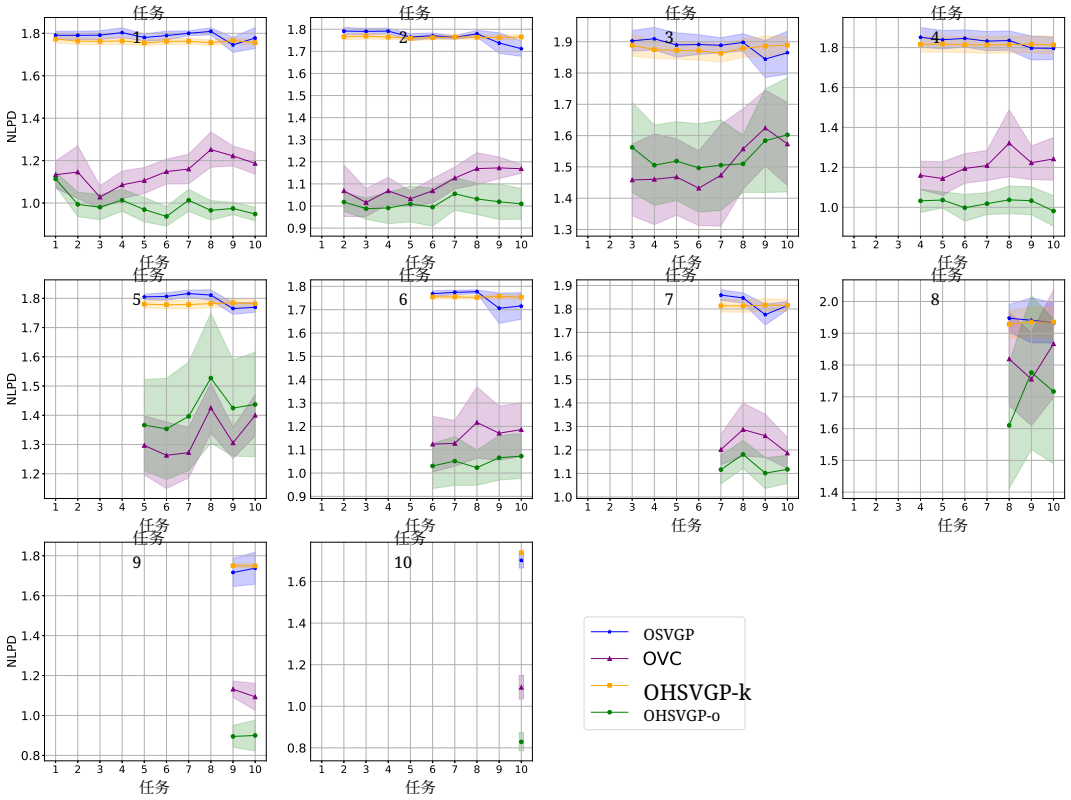
	太阳辐照度		**音频数据**		**COVID**	
方法	M		M		M	
	50	150	100	200	15	30
OSGPR/OSVGP	140	149	144	199	525	530
OVC	0.450	0.620	0.558	0.863	345	360
OVFF	0.327	0.354	0.295	0.356	-	-
OHSGPR/OHSVGP	0.297	0.394	0.392	0.655	370	380

可以获得闭式后验分布时，在考虑固定核超参数的情况下，OHSGPR 完全无需训练。因此，OHSGPR、OVC 和 OVFF 的运行速度显著更快，对于太阳辐照度和音频数据，可在数秒内适应全部任务。对于 COVID 数据，即使使用未折叠证据下界学习诱导变量的自由形式变分参数，由于无需对诱导位置计算梯度，OHSVGP 和 OVC 仍然显著快于 OSVGP。

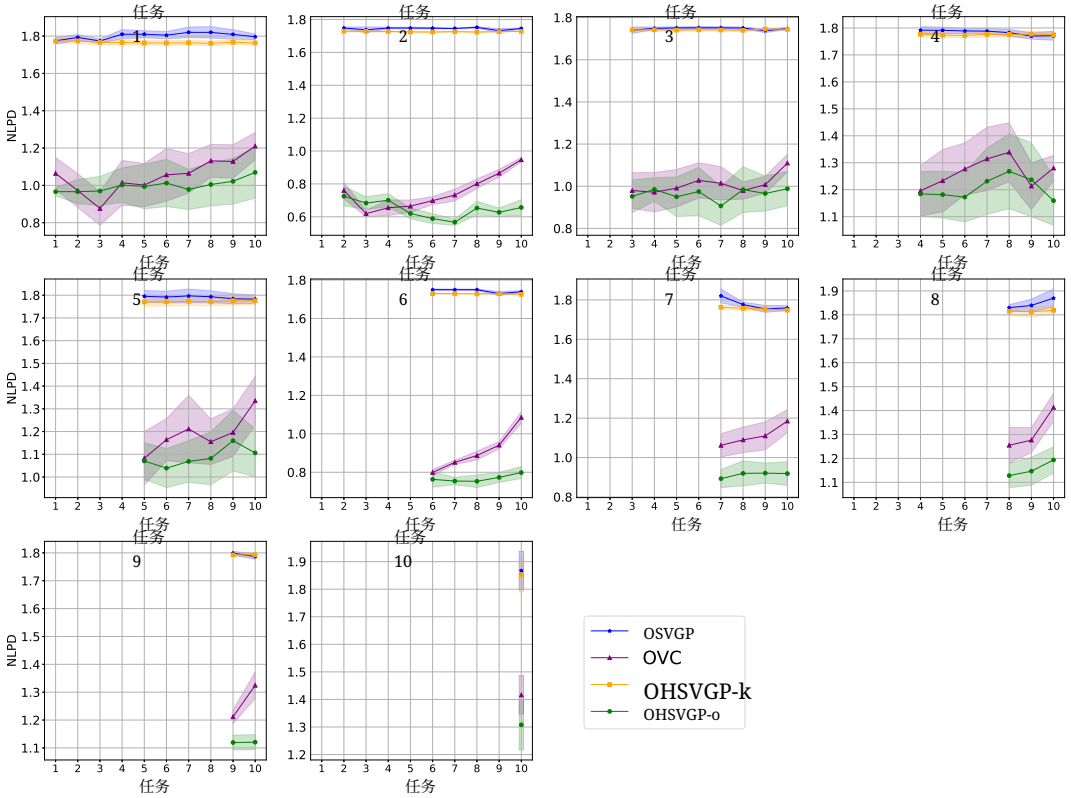
4.6.2 UCI 数据集上的持续学习

所有方法均使用 256 个诱导变量；对于每个任务，我们以 0.005 的学习率训练每种方法 2000 次迭代。这里我们只考虑 OVC，因为初步试验表明，在这两个数据集上，OVC-optZ 的结果较差。如第 4.4 节所述，在每个任务内，OHSVGP 需要对数据

点进行排序，以通过递推计算先验协方差矩阵。我们考虑两种排序准则。第一种称为 OHSVGP-o，它采用与任务创建方式相符的预言机顺序（例如，如果任务最初是根据与原点的 L2距离划分的，则按该距离排序）。在现实问题中，我们通常无法获知分布如何随任务变化。因此，我们还考虑 OHSVGP-k（与第 4.4 节中的 OHSVGP-k-max 相同），它采用基于核相似度的启发式排序方法：在任务j中，我们将第 i 个点选为满足 $\boldsymbol{x}$ 与 $\boldsymbol{x}_{i-1}^{(j)}$ 条件的 $\boldsymbol{x}_i^{(j)} = \arg\max_{\boldsymbol{x}\in\boldsymbol{X}^{(j)}} k(\boldsymbol{x}, \boldsymbol{x}_{i-1}^{(j)})$（对于 $i > 1$）；并将第一个任务中的第一个点设为 $\boldsymbol{x}_1^{(1)} = \arg\max_{\boldsymbol{x}\in\boldsymbol{X}^{(1)}} k(\boldsymbol{x}, 0)$。图 4.8 至图 4.11 对比了 OHSVGP 的两种变体与 OSVGP 和 OVC。对于所有 i，我们报告了在学习任务 $j = i$、$i + 1$、…、10 后，对任务 i 的评估结果（以 NLPD 和均方根误差表示）。总体而言，OSVGP 的表现最差，而且再次容易遗忘较早的任务，在图 4.10a 中尤为明显。OVC 在 Skillcraft 上表现尚可，但在图 4.10a 中也表现出灾难性遗忘。虽然 OHSVGP-k 在 Skillcraft 上的表现与 OSVGP 相近，但 OHSVGP-o 在全部 4 种场景中始终优于其他方法，这表明在将 OHSVGP 用于持续学习时，采用合理的排序方法至关重要。

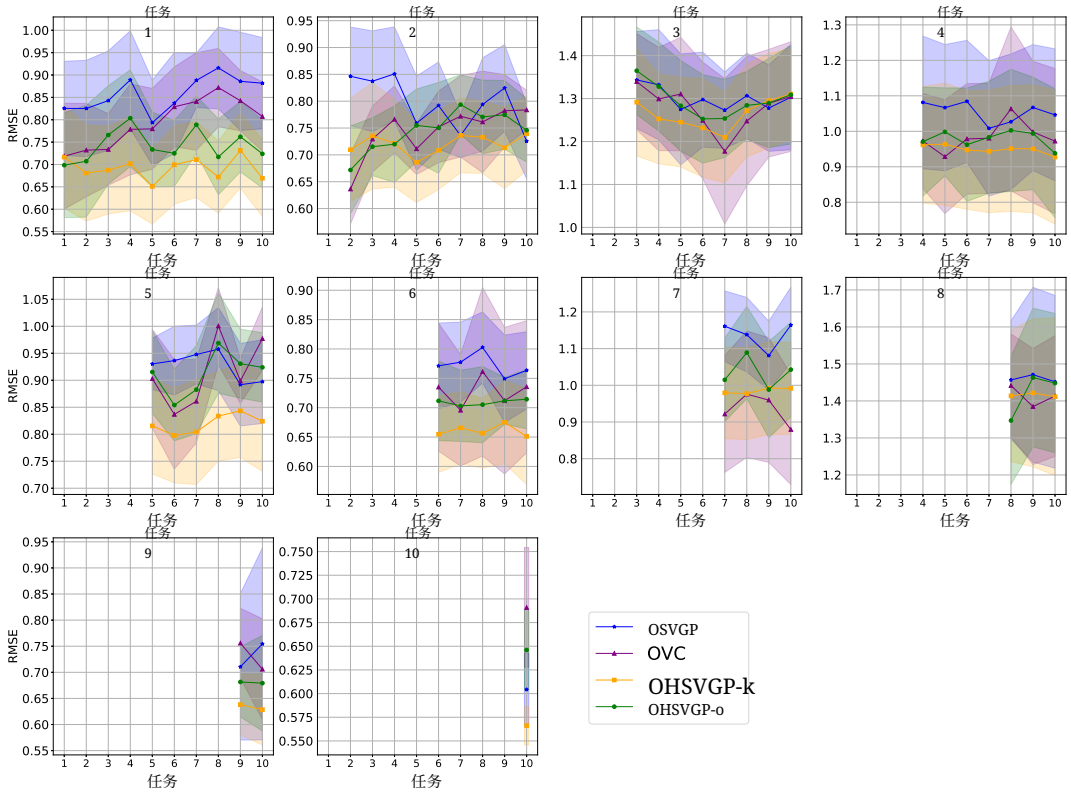


(a) Skillcraft（第1维度）

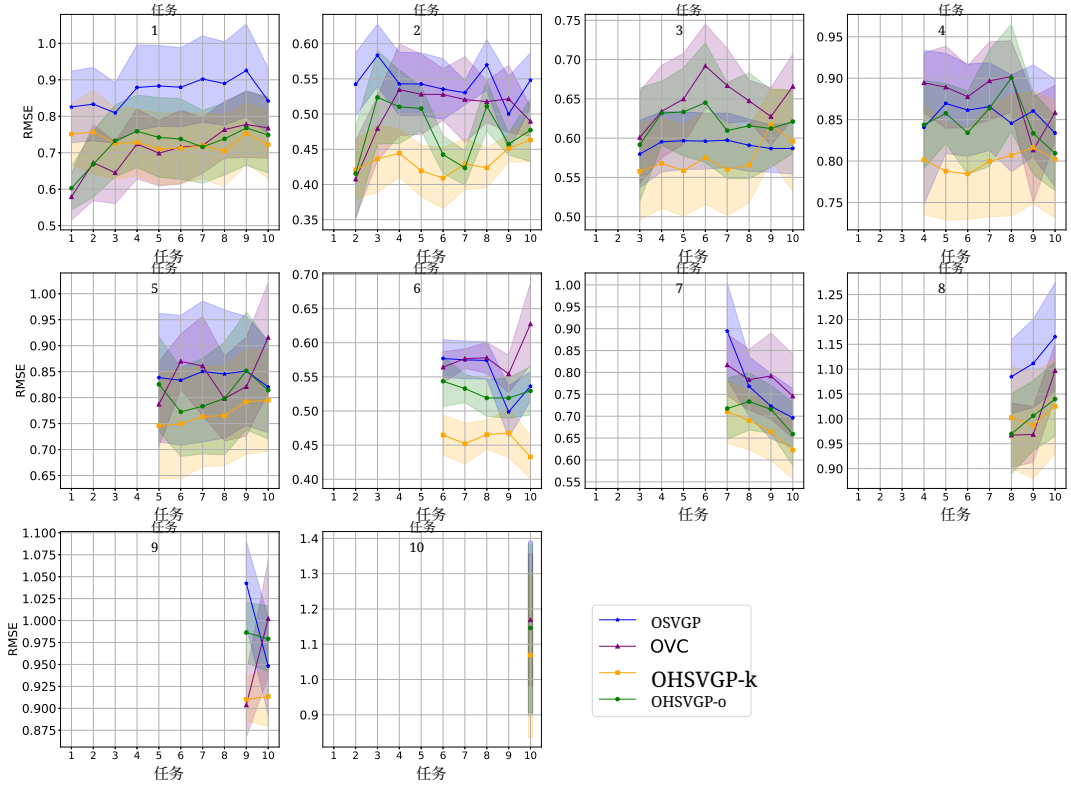


(b) Skillcraft（L2）

图 4.8：在 UCI Skillcraft 数据集的全部 10 个任务上持续学习每个任务后，各任务的测试集负对数预测密度。

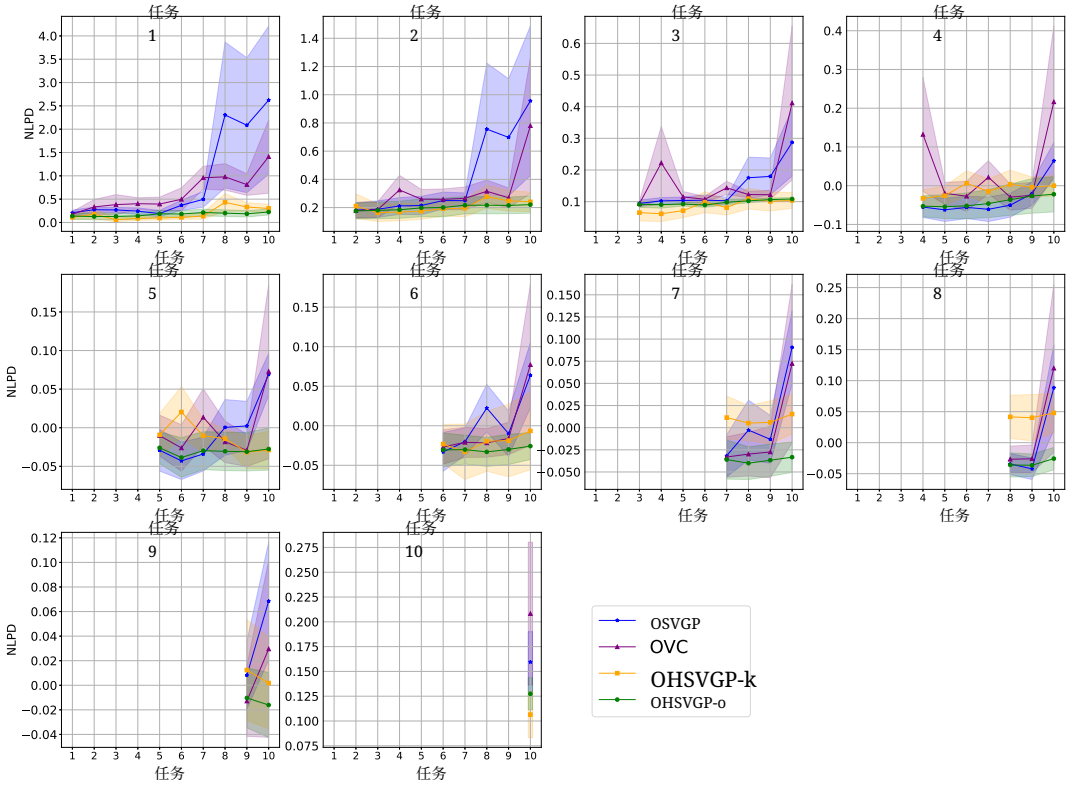


(a) Skillcraft（第 1 维度）

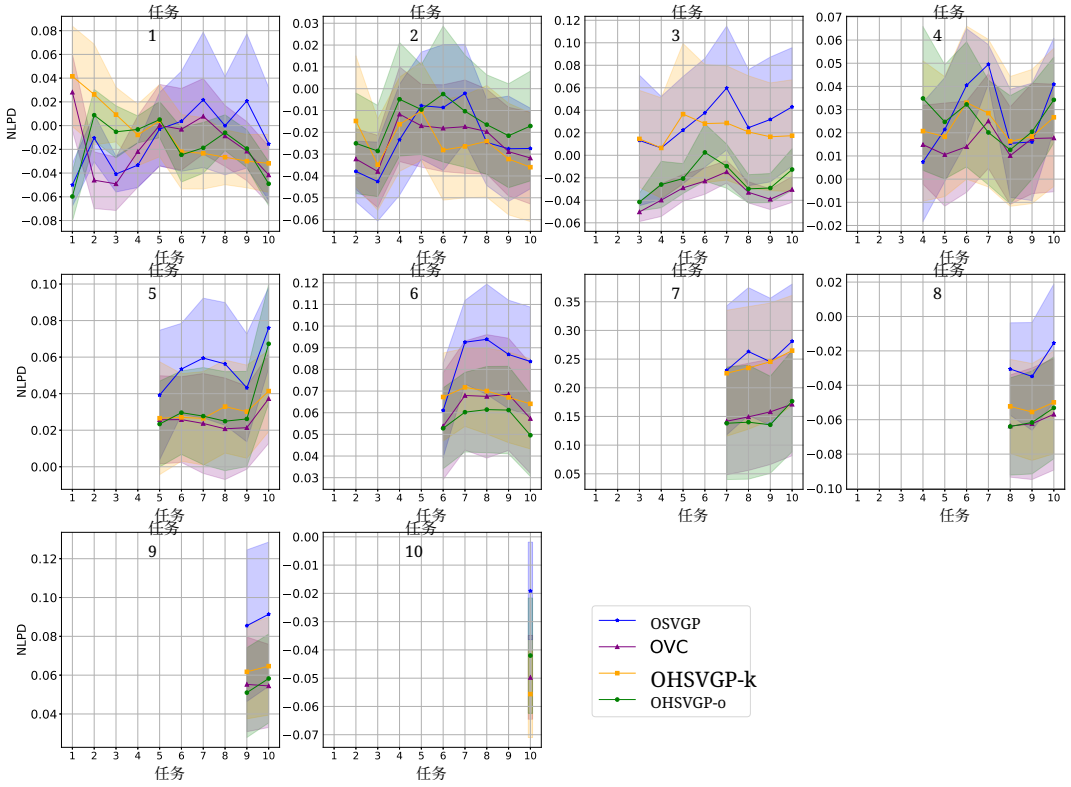


(b) Skillcraft（L2）

图 4.9：在 UCI Skillcraft 数据集上持续学习全部 10 个任务后，每个任务的测试集均方根误差。

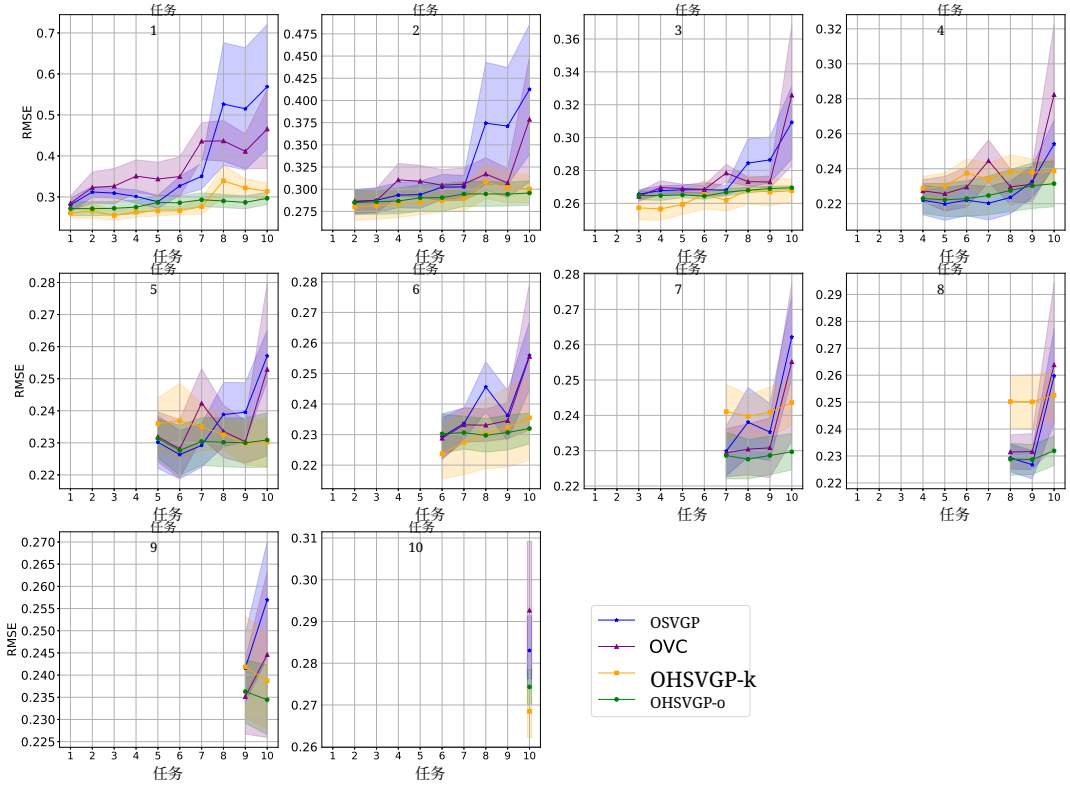


(a) 发电厂（第 1 维度）

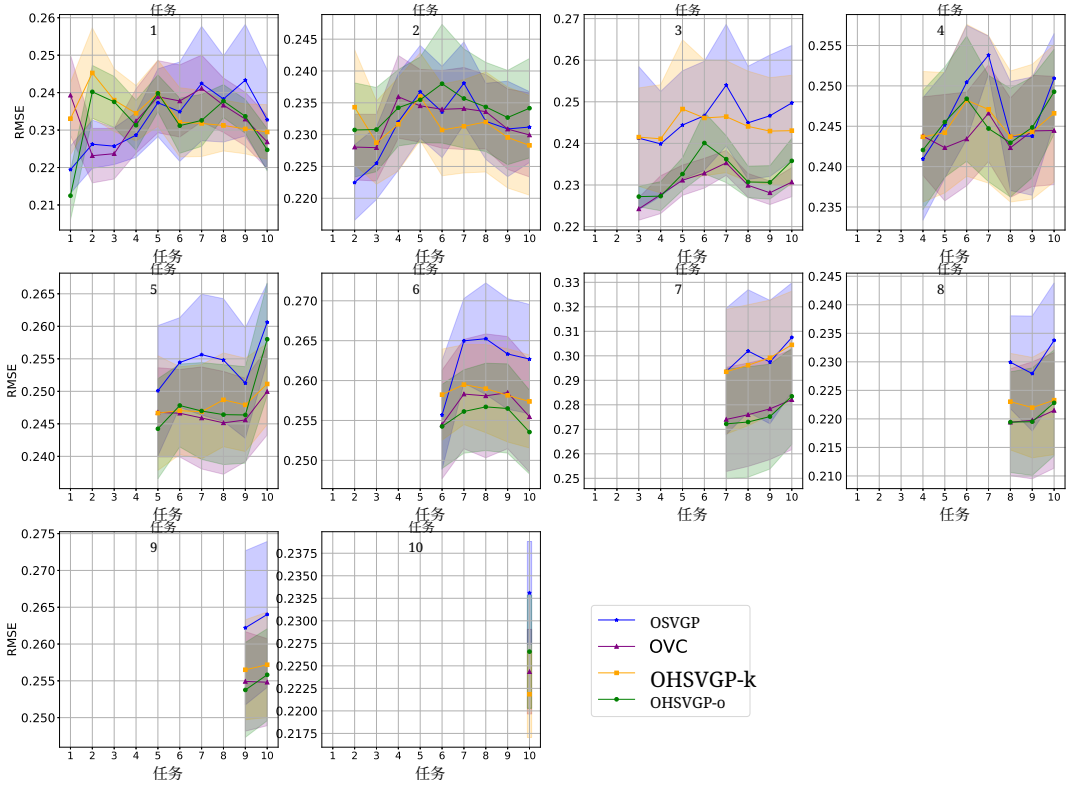


(b) 发电厂（L2）

图 4.10：在 UCI 发电厂数据集上持续学习全部 10 个任务后，各任务的测试集负对数预测密度。



(a) 发电厂（第 1 个维度）

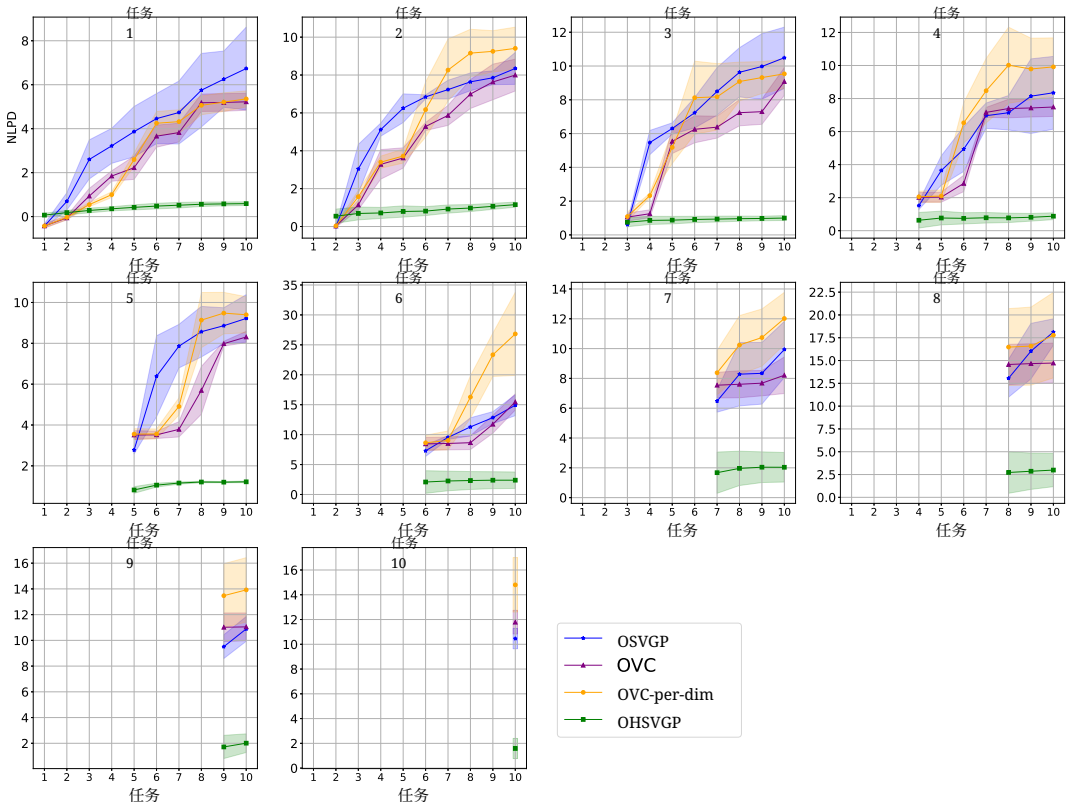


(b) 发电厂（L2）

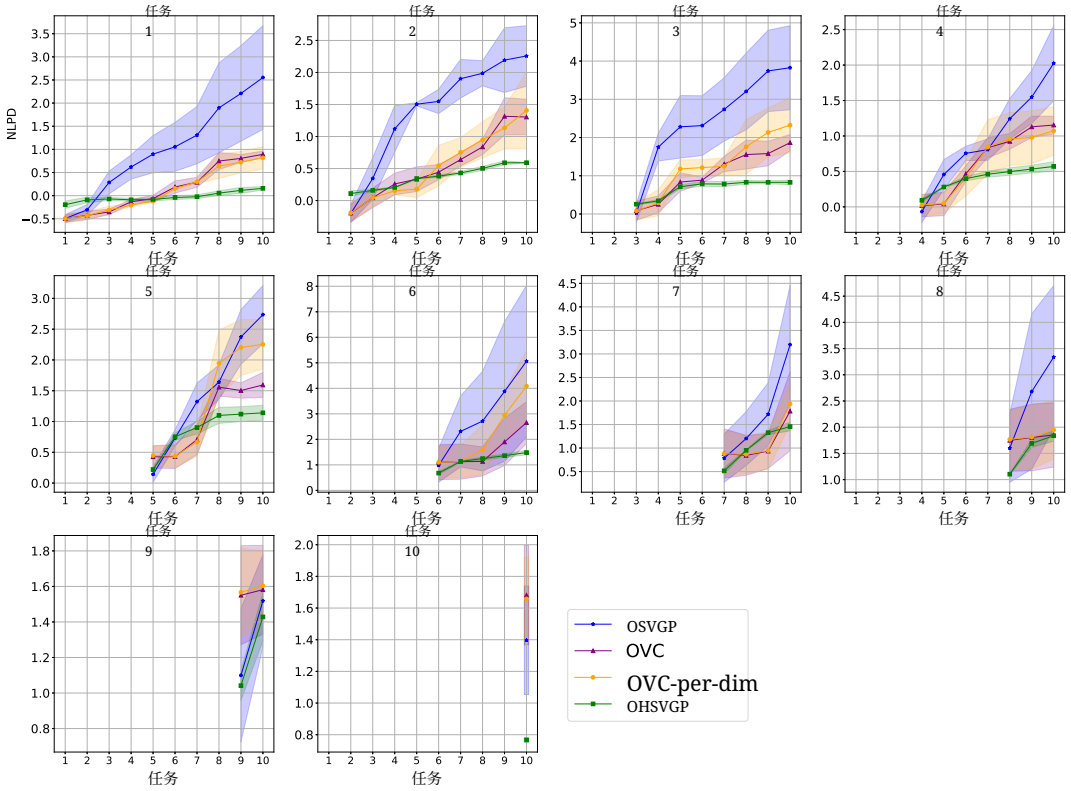
图 4.11：在 UCI 发电厂数据集上持续学习全部 10 个任务后，各任务测试集的均方根误差。

4.6.3 高维时间序列预测的持续学习

所有模型均采用相同的两层多层感知机编码器-解码器、20维潜在空间，以及由独立分量构成的多输出高斯过程；我们使用 $M \in \{50,100\}$，并在单个 NVIDIA A6000 GPU 上训练每个任务 20 个训练轮次，学习率为 0.005。SVGPVAE 中的持续学习通过在编码器和解码器上进一步施加弹性权重巩固（EWC；（Kirkpatrick 等人，2017年））损失来实现，由此得到朴素基线模型在线 SVGPVAE（OSVGP）。由于仅使用 EWC 不会对诱导位置进行正则化，因此，采用有原则的在线诱导位置放置规则可以改进模型。为此，我们还考虑了 OVC-SVGPVAE（OVC），它通过主元 Cholesky 分解在线调整诱导位置；以及按维度划分的 OVC-SVGPVAE（OVC-per-dim），它为每个潜在维度分配一组独立的 M 诱导点，从而提高 OVC 的灵活性。我们的方法在线 HiPPO SVGPVAE（OHSVGP）以 HiPPO 诱导变量替代 SVGPVAE 中的标准诱导点，并通过递推在线更新这些变量。图 4.12 和图 4.13 分别绘出了持续学习过程中所有任务的负对数预测密度和均方根误差的变化。OHSVGP 的性能始终保持稳定，而其他方法均表现出明显的灾难性遗忘：学习当前任务 i 后与学习最终任务 10 后的性能差距很大。造成这一差距的原因可能有两个：第一，标准诱导点无法充分覆盖较长的时间轴，而 OHSVGP 将诱导变量与基函数而非时间位置关联；第二，增加的编码器-解码器复杂度使必须重复使用有限诱导集的模型更难优化。增大 M 可以缩小这一差距，但计算开销和$\mathcal{O}(M^3)$，内存开销则为$\mathcal{O}(M^2)$，这凸显了 OHSVGP 更高的效率。

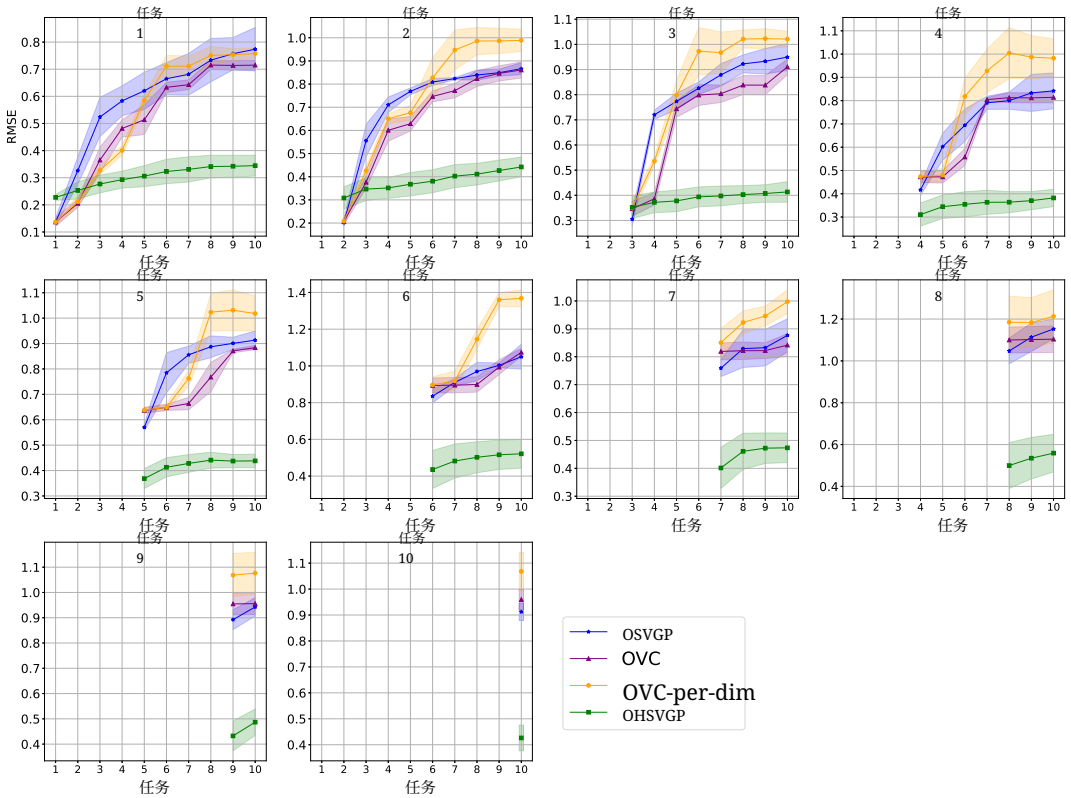


(a) M=50

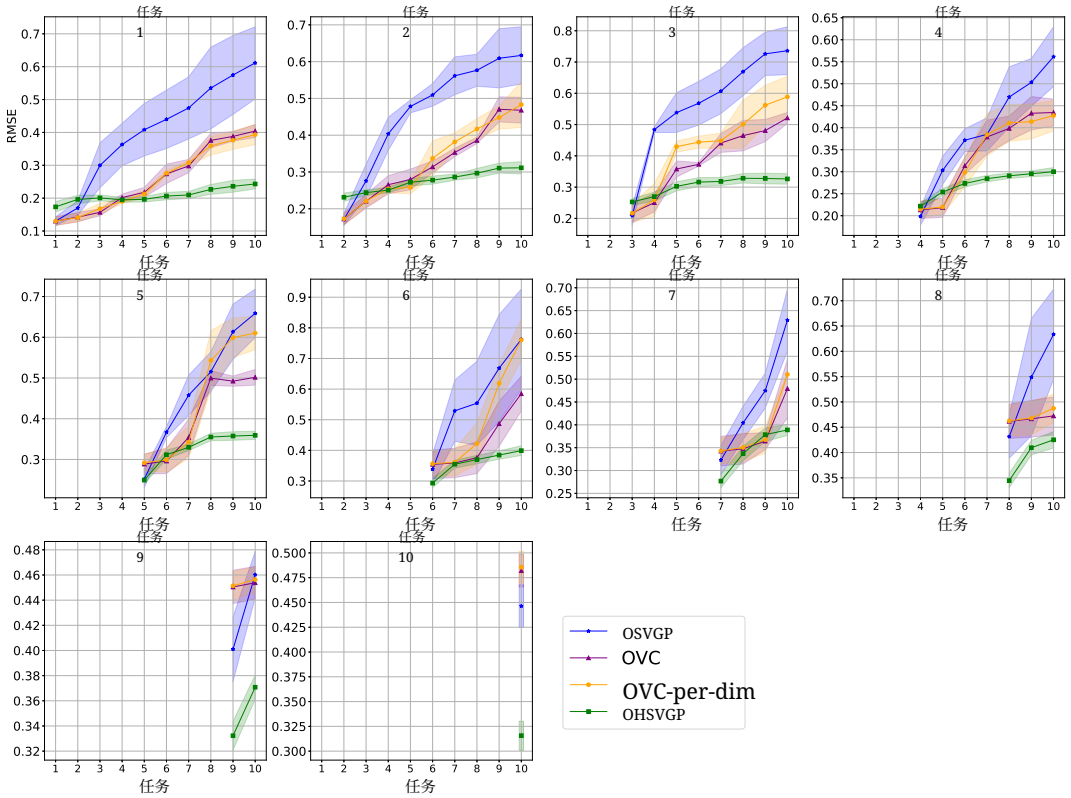


(b) M=100

图 4.12：在 ERA5 数据集上持续学习全部 10 个任务后，各任务的测试集负对数预测密度。



(a) M=50



(b) M=100

图 4.13：在 ERA5 数据集的全部 10 个任务上逐一持续学习后，各任务的测试集均方根误差。

4.7 结论

我们提出了 OHSVGP，这是一种新颖的在线高斯过程模型，利用 HiPPO框架为在线/持续学习提供稳健的长程记忆。通过将 HiPPO 的时变正交投影解释为自适应域间高斯过程基函数，我们利用状态空间模型改进在线高斯过程。这种联系使 OHSVGP 能够发挥 HiPPO 基于常微分方程的高效递归更新优势，同时保留基于高斯过程的不确定性感知预测。在一系列在线学习和持续学习任务上的实证结果表明，OHSVGP 的表现优于现有的在线稀疏高斯过程方法，尤其是在需要长期记忆的场景中。此外，其基于递推的协方差更新所产生的计算开销远低于 OSVGP 的顺序诱导点优化。这种高效的流式处理能力以及对历史信息的保留，使 OHSVGP 非常适用于既要求速度又要求准确率的现实应用。

第5章

你的图像其实是伪视频的最后一帧

本章基于Chen 等人(2025a)：

- Wenlong Chen*、Wenlin Chen*、Lapo Rastrelli 和 Yingzhen Li。**你的图像其实是伪视频的最后一帧**。发表于*ICLR* 机器学习深度生成模型：理论、原理与效能（*DeLTa*）研讨会，2025 年。主要构想由我提出，末位作者提供了有益的建议。两位共同第一作者（*）在末位作者的指导下，参与了代码实现、实验和论文撰写。第三位作者协助进行了与视频扩散模型相关的实验。我还协助统筹了其他作者的贡献。

在前两章中，我们结合深度序列建模和概率方法，构建了强大的预测模型。本章将探讨深度生成模型这一正交领域中的序列建模。我们的关键观察是，两类序列生成模型——扩散模型和分层变分编码器——之间的主要差异之一，在于扩散模型将归纳偏置作为直接监督信号，应用于其中间潜变量序列。受这一观察及扩散模型在经验上的成功启发，我们提出反复对图像进行数据增强，以创建伪视频序列，并探索通过联合建模由原始图像及其包含自监督信息的对应伪视频构成的序列分布，来改进其他图像生成模型的可能性。

5.1 引言

序列模型是一种广受欢迎的图像生成框架（Gulrajani 等人，2017年；Sønderby 等人，2016年；Ho 等人，2020年；Liu 等人，2022年；Albergo 等人，2023年；Lipman 等人，2023年；Shi 等人，2023年；Wang 等人，2024年）。这些模型并非一次性从噪声中生成图像（这可能颇具挑战），而是通过多个中间步骤，逐步将噪声转换为图像。近年来，其中的扩散模型（Sohl-Dickstein 等人，2015年；Ho 等人，2020年；Song 等人，2021b）及其变体（Kingma 等人，2021年；Nichol 和 Dhariwal，2021年；Song 等人，2021a；Rissanen 等人，2023年；Bansal 等人，2023年；Hoogeboom 和 Salimans，2023年）展现出令人瞩目的高质量图像生成能力。

虽然扩散模型可以看作传统序列生成模型的一种特殊情况，即层次变分自编码器（HVAEs）（Sønderby 等人，2016年；Maaløe 等人，2019年；Vahdat 和 Kautz，2020年），但在实际应用中，它们的表现往往显著优于标准 HVAEs。扩散模型与标准 HVAEs 的主要区别有两点。首先，扩散模型通常包含更多中间状态，这或许有助于提升生成质量（Huang 等人，2021年）。其次，扩散模型为其中间状态提供了确切的自监督信息：这些状态应当与原始目标图像在不同受损程度下受损（例如带噪或模糊）的版本相匹配。这些额外信息有助于规范训练，并指导扩散模型的生成过程。另一方面，标准 HVAEs 中的中间状态不可观测，因此无法对其进行显式控制。因此，可能存在许多不同的中间状态分布，都能够生成图像（即不可识别性问题（Locatello 等人，2019年；Khemakhem 等人，2020年））。中间状态缺乏可识别性可能会给训练过程中的优化带来挑战，因为这意味着假设空间极其庞大，其中存在许多次优解。

在本文中，我们提出假设：将此类自监督信息融入扩散模型等灵活的生成模型中，可能是其取得良好生成性能的关键原因之一。基于这一假设，我们探索了通过将其他类型的图像生成模型扩展为视频生成模型，并以伪视频的形式人为注入自监督信息，从而改进这些模型的可能性。这些伪视频的帧是通过对原始图像应用数据增强生成的。随后，我们使用这些伪视频训练视频生成模型。之后，我们将视频生成模型生成的伪视频末帧（对应于原始图像）的生成质量，与原始图像

生成模型直接生成的图像进行比较。通过实验证实，与原始图像生成模型直接生成的图像相比，采用伪视频生成能够提升图像生成质量。从理论上看，我们阐释了为什么利用数据增强设计超越一阶马尔可夫链的更优伪视频可能会有所帮助。

本文的贡献总结如下。

- （第 5.2节）我们的核心洞见是，通过对原始目标图像施加扰动所创建的伪视频，可能为训练生成模型提供有用的自监督信息。我们通过比较扩散模型与标准分层变分自编码器，将其作为动机示例加以论证。

- （第 5.3 节和 5.4节）我们尝试改进两个广泛使用的生成模型框架，即 VQVAE（Van Den Oord 等人，2017年）和改进型 DDPM（Nichol 和 Dhariwal，2021年），方法是将它们扩展为相应的视频生成模型，并在伪视频上进行训练。通过实验证明，仅使用包含少量帧的伪视频，该过程就能提升图像生成质量。总体而言，我们提出的框架为利用相应的视频生成模型扩展图像生成模型、从而获得潜在性能提升提供了一种新方法。

- （第 5.4节）在理论上，我们分析了自回归视频生成框架中某些伪视频（包括以一阶马尔可夫链形式存在的伪视频）可能带来的问题。基于理论结果，我们提出了一种简单而有效的方法，通过构建高阶马尔可夫伪视频来避免这些潜在问题。

5.2 动机

设 $\boldsymbol{x} \in \mathbb{R}^n$ （亦记为 $\boldsymbol{z}_0$）为所关注的观测变量。生成建模的任务旨在拟合参数化模型 $p_\theta(\boldsymbol{x})$，以利用来自 $p(\boldsymbol{x})$ 的样本估计数据分布 $p(\boldsymbol{x})$。我们已分别在第 2.4.2 节和第 2.4.3 节回顾了层次变分自编码器（分层变分自编码器；(Sønderby 等人，2016；Maaløe 等人，2019；Vahdat 和 Kautz，2020)）和扩散模型（Sohl-Dickstein 等人，2015；Ho 等人，2020；Song 等人，2021b）；若需重温相关背景与记号，请读者参阅这两节。我们的关键观测来自这两类深度序列生成模型的训练目标之间的差异：

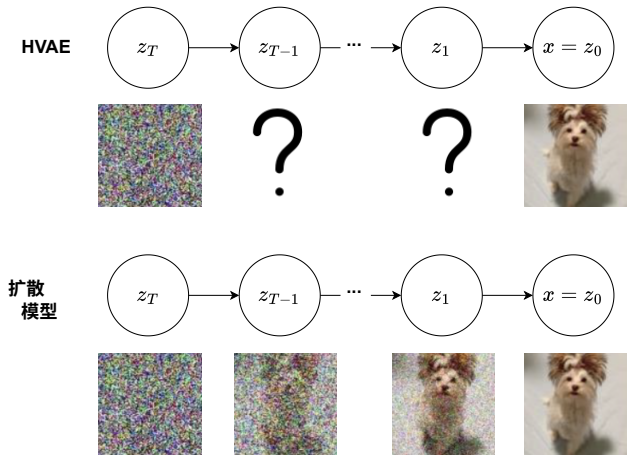


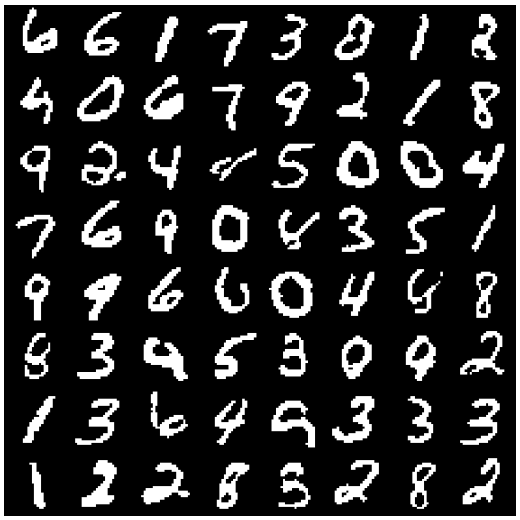
图 5.1：扩散模型与 HVAE：与标准分层变分自编码器相比，扩散模型通过自监督信号为其中间潜在状态融入了归纳偏置。

$$\mathcal{L}_{ELBO}^{\text{HVAE}}(\boldsymbol{x}, \boldsymbol{\theta}, q_{\boldsymbol{\phi}}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0)}\left[\log \frac{p(\boldsymbol{z}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0)}\right], \tag{5.1}$$

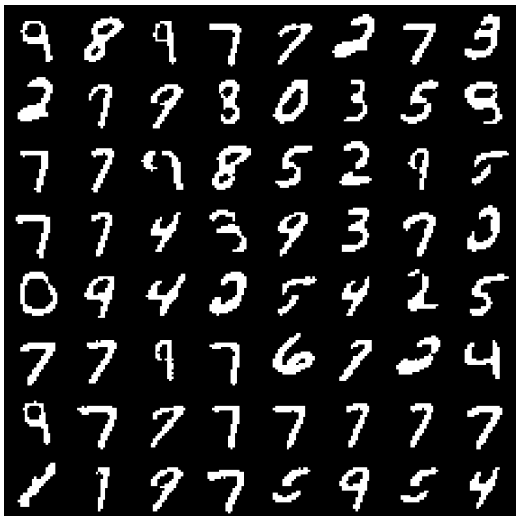
$$\begin{aligned}\mathcal{L}_{ELBO}^{\text{Diffusion}}(\boldsymbol{x}, \boldsymbol{\theta}) &= \mathbb{E}_{q(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_0|\boldsymbol{z}_1) - \sum_{t=1}^{T} \mathrm{KL}(q(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{z}_0)||p_{\boldsymbol{\theta}}(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}))\right] \\ &= \mathbb{E}_{q(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}_0|\boldsymbol{z}_1) - \sum_{t=1}^{T} \underbrace{\frac{\|\mu_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t) - \tilde{\mu}(\boldsymbol{z}_t, \boldsymbol{z}_0)\|^2}{2\sigma_t^2}}_{\text{mean matching}}\right].\end{aligned} \tag{5.2}$$

5.2.1 扩散模型与分层变分自编码器

与训练标准分层变分自编码器的目标函数（式 5.1）相比，可以看出，训练扩散模型的目标函数（式 5.2）纳入了对中间状态的直接控制。由于推断模型是固定且预先定义的（式 2.58），式 5.2 中的目标函数得以简化。具体而言，其第二项（均值匹配）表明，在每个中间步骤 t，生成模型中的均值函数 $\mu_\theta(\boldsymbol{z}_t, t)$ 通过匹配加噪版本 $\tilde{\mu}(\boldsymbol{z}_t, \boldsymbol{z}_0)$（即原始目标图像$\boldsymbol{z}_0$）来进行训练：$= \boldsymbol{x}$。相比之下，标准分层变分自编码器的目标函数并不包含此类信息，因而无法对其中间状态施加任何控制，因为其推断模型由灵活的神经网络参数化，并随生成模型通过端到端训练持续更新。图 5.1 说明了这一差异



(a) 采用热方程编码器的分层变分自编码器



(b) 采用可学习编-
码器

图 5.2：采用根据热方程固定的编码器的分层变分自编码器，以及采用可学习编码器的标准分层变分自编码器所生成的数字图像。两个分层变分自编码器均使用与Rissanen 等人(2023年)中相同的解码器架构

扩散模型与标准分层变分自编码器之间的差异。在没有自监督信息的帮助下，标准分层变分自编码器中的中间状态具有很大的灵活性，这意味着从不可识别性的意义上说，可能存在许多关于这些中间状态的合理分布，而这些分布都能够生成图像（即许多次优解）；随着 T 变大，这会使优化更加困难。相比之下，扩散模型的中间状态可能受益于自监督信息（即噪声图像），因此即使 T 较大，优化也可能不那么困难，因为这种归纳偏置确定了一条特定的生成路径，从而排除了与该归纳偏置不一致的其他解。

表 5.1：采用依据热方程固定的编码器的 HVAE，以及采用可学习编码器的标准 HVAE 所生成数字的 Inception分数（IS）。

	采用热方程编码器的 HVAE	采用可学习编码器的标准 HVAE
IS	**9.32**	7.27

为了展示自监督信息的关键作用，我们在二值化 MNIST 数据集（Salimans 等人，2015年）上训练了一个架构与 Rissanen 等人 等人2023年）相似的 HVAE，作为概念验证；其中编码器依据热方程固定（Rissanen 等人，2023年）：

$$q(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0) = \prod_{t=1}^{T} q(\boldsymbol{z}_t|\boldsymbol{z}_0) = \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{z}_t|F_h(t)\boldsymbol{z}_0, \sigma_h^2), \tag{5.3}$$

whe其中 F_h 是用于模拟热方程直至时间 t 的矩阵。这可 n

视为一个 HVAE，它利用由热方程生成的伪视频提供的显式监督信号进行训练。我们为每张训练图像创建 $T = 18$ 帧伪视频。图 5.2 展示了，使用热方程生成的伪视频训练的 HVAE，相较于采用相同架构但使用可学习编码器的标准 HVAE，能够生成清晰得多、种类也更丰富的数字。我们还在表 5.1 中报告了生成图像的Inception分数（IS），以便进行定量比较。使用热方程生成的伪视频训练的 HVAE，其 IS 明显高于标准 HVAE。在此，我们特意使用热方程而非高斯噪声来创建伪帧，以说明除了像标准扩散模型那样添加高斯噪声之外，还有许多方法可以创建包含有用自监督信息的伪视频。

5.2.2 通过伪视频生成改进图像生成

鉴于伪视频这一概念及其在扩散模型中的有效性，我们在本文中关注以下尚未解决的一般性问题：

> 能否通过联合建模原始图像的分布及其对应的、包含自监督信息的伪视频分布，来改进其他类型的图像生成模型？

答案是肯定的。本文展示了伪视频在两类生成模型上的优势的实证证据：分别利用在伪视频上训练的 Phenaki（Van Den Oord 等人，2017年）（第 5.3 节）和视频扩散（Nichol 和 Dhariwal，2021年）（第 5.4.2 节），改进 VQVAE（Villegas 等人，2022年）和 DDPM（Harvey 等人，2022年）。此外，我们还从理论上论证，在自回归视频生成框架中，应采用表达能力更强的伪视频构造方式（第 5.4.1 节），而不应局限于标准扩散模型前向过程中通常采用的一阶马尔可夫策略。

5.3 利用伪视频改进 VQ-VAE 的重建与生成

在本节中，我们利用伪视频来提升向量量化变分自编码器（VQVAE）的图像生成质量（Van Den Oord 等人，2017年），其中潜变量 z 是离散标记。

5.3.1 VQVAE 的预备知识

向量量化变分自编码器（VQVAE）与变分自编码器（VAE）类似，也是一种深度潜变量模型，假设数据生成过程的形式如下：

$$\boldsymbol{x} \sim \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z}. \tag{5.4}$$

VQVAE 与 VAE 有两点不同。首先，VAE 假设 $\boldsymbol{z}$是连续的，并假设 $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ 先验服从标准高斯分布；而 VQVAE 使用离散潜变量 $z \in \{1, \cdots K\}$，并通过另一个深度神经网络对 $p_{\boldsymbol{\theta}}(z)$ 进行参数化。其次，VAE 通过近似最大似然估计来训练与似然 $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ 相关的生成模型参数$\boldsymbol{\theta}$ ，而 VQVAE 则采用两步训练流程，分别拟合似然 $p_{\boldsymbol{\theta}}(\boldsymbol{x}|z)$ 和先验 $p_{\boldsymbol{\theta}}(z)$。给定观测训练集 $\{\boldsymbol{x}_n\}_{n=1}^N$，第一步是学习条件分布 $p_{\boldsymbol{\theta}}(\boldsymbol{x}|z)$，通过辅助编码器$z_{\boldsymbol{\phi}}(\boldsymbol{x}) : \mathcal{X} \to \{1, \cdots K\}$来实现（与 VAE 中推断模型的概念类似）：

$$z_{\boldsymbol{\phi}}(\boldsymbol{x}) = \underset{i \in \{1, \cdots, K\}}{\arg\min} ||f_{\boldsymbol{\phi}}(\boldsymbol{x}) - \boldsymbol{z}_i^{\text{emb}}||, \tag{5.5}$$

其中， $f_\phi(\cdot) : \mathcal{X} \to \mathbb{R}^d$ 由神经网络参数化， $\boldsymbol{z}_i^{\text{emb}} \in \mathbb{R}^d$ 是与离散编码 i相关的d维潜在嵌入。随后，我们针对每个观测，使用以下损失函数联合训练编码器和似然：

$$\begin{aligned} \mathcal{L}_{VQVAE}(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\phi}, \{\boldsymbol{z}_i\}_{i=1}^K) = &- \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|k) + ||sg(f_{\boldsymbol{\phi}}(\boldsymbol{x})) - \boldsymbol{z}_k^{\text{emb}}||_2^2 \\ &+ \beta ||f_{\boldsymbol{\phi}}(\boldsymbol{x}) - sg(\boldsymbol{z}_k^{\text{emb}})||_2^2, \quad k = z_{\boldsymbol{\phi}}(\boldsymbol{x}) \end{aligned} \tag{5.6}$$

其中 $sg(\cdot)$表示“停止梯度”算子。目标函数的第一项是负对数似然，用于鼓励准确重建；第二项是量化损失，确保潜在嵌入接近编码器网络输出 $f_{\boldsymbol{\phi}}(\boldsymbol{x})$；第三项是承诺损失，确保当 $\beta < 1$时编码器参数的更新不会过快，从而使编码器能够承诺于某一嵌入。

请注意，潜在空间可以由多维离散标记表示。例如，对于RGB图像 $\boldsymbol{x}$ ，其大小为 $H \times W \times 3$， $f_\phi(\cdot)$通常由卷积神经网络参数化，该网络将 $\boldsymbol{x}$ 映射到大小为 $h \times w \times d$ 的嵌入，并且 $f_{\boldsymbol{\phi}}(\boldsymbol{x})$的每个维度被量化为 K 个离散标记之一，如式5.5所示，因此 $\boldsymbol{z}_{\boldsymbol{\phi}}(\boldsymbol{x})$是大小为 $h \times w$ 、含离散元素的张量。

在学习的第二步 $p_{\boldsymbol{\theta}}(\boldsymbol{z})$中，我们将训练集中的每个 $\boldsymbol{x}$ 映射为潜在离散词元，从而形成离散潜变量的经验分布，

$p(\boldsymbol{z}) \approx p_{\text{empirical}}(\boldsymbol{z}) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\boldsymbol{z}_\phi(\boldsymbol{x}_n)}(\boldsymbol{z})$。接下来，我们可以为 $p_{\text{empirical}}(\boldsymbol{z})$选择一种特定的因子分解顺序，并拟合自回归模型（van den Oord 等人，2016a,b） $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t|\boldsymbol{z}_1, \cdots \boldsymbol{z}_{t-1})$，通过最大似然训练，使其根据此前生成的词元序列预测下一个离散编码；训练集取自$p_{\text{empirical}}(\boldsymbol{z})$。最终的先验模型为 $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}_1)p_{\boldsymbol{\theta}}(\boldsymbol{z}_2|\boldsymbol{z}_1)\cdots(\boldsymbol{z}_T|\boldsymbol{z}_{t<T_z})$，其中 T_z 是 $\boldsymbol{z}$ 中的离散词元总数（例如上述图像生成示例中的 $T_z = h \times w$）。

5.3.2 实验

为了将伪视频序列纳入模型，我们采用了 VQVAE 的视频生成模型版本 C-ViViT（Villegas 等人，2022年），将尺寸为 $T \times H \times W \times 3$ 的伪视频压缩为尺寸为 $t \times h \times w$的潜在离散词元。C-ViViT 不采用卷积神经网络，而是包含时间变换器层来处理时间维度；在其隐藏层中，高度和宽度维度被展平为大小为 $h \times w$的单一“空间维度”，并且 C-ViViT 还采用空间变换器层来计算沿空间维度的注意力。除了式 5.6中定义的标准 VQVAE 损失外，C-ViViT 还在最终目标函数中加入了 GAN 风格的对抗损失（Karras 等人，2020年）和图像感知损失（Johnson 等人，2016年；Zhang 等人，2018b）。对于 C-ViViT 创建的潜在空间，我们考虑两种生成模型来拟合潜在词元的先验 $p(\boldsymbol{z})$：

- VideoGPT（Yan 等人，2021年）使用自回归（AR）Transformer（Brown 等人，2020年）以自回归方式分解 $p(\boldsymbol{z}) = \prod_{i=1}^{T_z} p(\boldsymbol{z}_i|\boldsymbol{z}_{<i})$，并通过掩码自注意力进行建模，其中 T_z 表示词元总数，训练时采用最大似然。VideoGPT 是从 ImageGPT（Chen 等人，2020a）扩展而来的对应视频生成模型。

- Phenaki（Villegas 等人，2022年）使用双向 Transformer（Vaswani 等人，2017年）一次性预测所有词元，而不是以自回归方式逐个预测。在每个训练步骤中，抽取一个掩码比例 $\gamma \in (0, 1)$，并通过根据未掩码词元预测掩码词元来训练模型。生成时，所有词元最初都被掩码，模型会同时预测所有词元。随后经过几步重新掩码和重新预测来逐步优化生成结果，并随着步骤推进逐渐降低掩码比例。Phenaki 是从 MaskGit（Chang 等人，2022年）扩展而来的对应视频生成模型。

我们比较了在伪数据上训练的视频生成模型所生成视频中末帧（对应原始图像）的生成质量

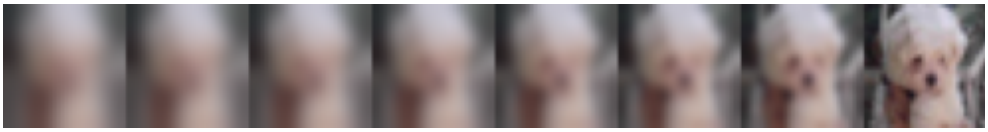


图 5.3：通过对狗的图像进行模糊处理构造伪视频的示例。

与在原始目标图像上训练的原始图像生成模型所生成的图像。下面我们列出实验设置，并在附录中提供详细的超参数D.2。

数据集。 我们使用两个基准数据集中的图像创建 8帧和 18帧伪视频：CIFAR10（32×32）（Krizhevsky 等人，2009年）和 CelebA（64×64）（Liu 等人，2015年），并采用 Bansal 等人（2023年）提出的模糊处理技术。创建 8帧伪视频时，我们使用大小为 11×11 的高斯核对图像进行 7 次递归模糊处理，标准差以 0.05 的速率呈指数增长。创建 18帧伪视频时，我们使用相同的高斯核对图像进行 17 次模糊处理，但标准差以 0.01 的速率呈指数增长。伪视频的排列方式是：最后帧为原始目标图像，第一帧为模糊程度最高的图像。图 5.3 展示了此类伪视频的一个示例。我们用 1帧表示由在原始目标图像上训练的原始图像 VQVAE 模型生成的图像。起初，我们也尝试使用高斯噪声进行数据增强，但未能取得理想结果。我们认为，这可能是因为 VQVAE 和 C-ViViT 会将数据压缩为离散潜在标记；若使用高斯噪声进行数据增强，每个图像块中像素值的变化就会增大。因此，要将变化如此之大的伪视频压缩为由有限码本大小定义的离散标记将十分困难。相比之下，模糊处理会降低图像中的变化，使彼此相邻的像素具有相似的值，从而让具有有限码本大小的 VQVAE 或 C-ViViT 更容易压缩伪视频。

网络架构。[1] 我们训练基于 VQVAE 的生成模型，离散潜在标记的码本大小（K）为 1024。具体而言，我们使用 C-ViViT 作为伪视频的压缩模型（自编码器）。对于形状为（T， H， W，3）的伪视频，我们通过提取大小为 $2 \times 4 \times 4$ 的视频块，将其压缩为形状为（$\frac{T}{2}$， $\frac{H}{4}$， $\frac{W}{4}$）的离散潜在标记。对于形状为（1， H，W，3）的图像，我们使用 VQVAE 通过提取大小为 4×4 的图像块，将其压缩为形状为（1， $\frac{H}{4}$， $\frac{W}{4}$）的词元。随后，我们考虑两种视频生成模型 VideoGPT 和

[1]对于1帧模型，我们尝试使用更深的架构，但未观察到性能提升，这表明我们的伪视频框架可以进一步提升生成模型的性能，而单纯增大模型规模则会失效。



(a) 真实值 (b) 1帧重建-构 (c) 8帧重建-tion (d) 18帧重建-构造

图 5.4：在 CIFAR10 上训练的 VQVAE/CViViT 生成的真实图像和重建图像。



(a) 真实值 (b) 1帧重建-构造 (c) 8帧重建图像tion (d) 18帧重建建

图 5.5：在 CelebA 上训练的 VQVAE/CViViT 的真实图像和重建图像。

Phenaki（以及对应的图像生成模型 ImageGPT 和 MaskGit），以拟合潜在标记的先验。

- **VQVAE/C-ViViT（重建）**。我们采用了与Villegas 等人(2022年）相似的架构，该架构包含一个 4 层空间 Transformer 和一个 4 层时间 Transformer，隐藏维度为 512。对于 1帧 VQVAE 模型，由于没有时间维度，我们采用 8 层空间 Transformer。

- **ImageGPT/VideoGPT（自回归生成）**。我们采用了与Yan 等人(2021年）相似的架构，该架构包含一个具有 4 个注意力头、隐藏维度为 144 的 8 层自回归（AR）Transformer。

- **MaskGit/Phenaki（潜在掩码生成）**。我们采用了与Villegas 等人（2022年）类似的架构，其中包含一个具有 512 隐藏维度的 6 层双向 Transformer。

评估指标。我们使用5万样本计算弗雷歇Inception距离（FID ↓）（Heusel等人，2017年），以评估图像的质量。这些图像来自在伪视频上训练的视频模型所生成视频的末帧，或

表 5.2：在基于 CIFAR10 和 CelebA 图像构建的伪视频上训练的 C-ViViT（重建）、VideoGPT（自回归生成）和 Phenaki（潜在掩码生成）所生成图像的最后一帧 FID/PSNR。1帧结果来自其图像对应模型 VQVAE（重建）、ImageGPT（自回归生成）和 MaskGit（潜在掩码生成），这些模型在原始 CIFAR10 和 CelebA 图像上训练。

	CIFAR10			CelebA		
	1帧	8帧	18帧	1帧	8帧	18帧
重建（FID）	24.53	13.81	**11.26**	24.62	5.72	**2.27**
重建（PSNR）	19.97	21.86	**25.89**	20.52	24.66	**29.68**
自回归生成（FID）	72.06	**54.60**	69.23	32.98	30.19	**28.08**
潜在掩码生成（FID）	49.27	**35.50**	47.65	27.34	16.87	**16.66**

来自在原始目标图像上训练的图像模型所生成的图像。我们还报告峰值信噪比（PSNR ↑），作为重建结果的补充指标。

结果。 表5.2 分别展示了 C-ViViT 在重建任务中的最后一帧 FID 和 PSNR，以及 VideoGPT 和 Phenaki 在 CIFAR10 和 CelebA 上进行自回归生成和掩码生成时的相应指标。1 帧的结果对应其图像模型对应版本的性能（即用于图像重建的 VQVAE，以及分别用于自回归生成和掩码图像生成的 ImageGPT 和 MaskGit）。我们发现，伪视频确实有助于改善 C-ViViT 的训练：增加少量帧后，末帧的重建质量显著提升。我们在图5.4 和 5.5中分别展示了不同 C-ViViT 模型重建的 CIFAR10 和 CelebA 图像。可以看出，对于这两个数据集，使用 1 帧模型训练得到的重建图像都较为平滑；而使用伪视频后，这一问题得以解决，生成的图像也更加清晰。从定量结果来看，随着使用的帧数增加，重建 FID 会有所改善。与 1 帧结果相比，伪视频也能提升图像生成性能。有趣的是，随着帧数增加，收益会逐渐递减。对于 CelebA 图像，18 帧伪视频在自回归生成和潜在掩码生成中都能取得最佳图像生成性能，但 8 帧模型的表现也与之相当。对于 CIFAR10 图像，8 帧伪视频的图像生成性能优于 18 帧伪视频，这表明要很好地重建包含 18 帧的伪视频，潜在标记需要具有更复杂的先验分布，因此 VideoGPT 或 Phenaki 更难捕捉这一先验。总之，虽然伪视频有助于提升性能，但最佳帧数可能取决于数据集和增强策略。在实践中，这种收益递减并非严重问题，因为从业者可以选择只增加少量伪帧来改善

生成效果，从而避免引入高昂的计算成本。

5.4 通过高阶马尔可夫伪视频改进生成

由于伪视频包含关于目标图像的额外信息，我们希望进一步了解可以利用哪些类型的额外信息来提升图像生成质量。在实践中，创建伪视频的数据增强策略有无数种，因此我们希望研究哪些类型的数据增强更有利，从而为伪视频的实际设计提供参考。

5.4.1 一阶马尔可夫链是创建伪视频的最佳选择吗?

考虑伪视频 $\boldsymbol{z}_{1:T}$，其中 $\boldsymbol{z}_T := \boldsymbol{x}$ 是目标图像[2]，而 $\boldsymbol{z}_t$'s $(t < T)$ 是通过某种数据增强生成的 $\boldsymbol{z}_T$ 的一些噪声测量。我们表明，原则上，利用更多伪帧来生成$\boldsymbol{z}_T$ 的生成模型更有可能取得更好的性能；而像标准扩散模型那样，通过一阶马尔可夫链将目标图像的信息传递给伪帧，未必是最佳选择。我们通过以下自回归视频生成示例来说明这一点。

考虑构建一个生成模型 g，该模型仅利用 $\boldsymbol{z}_{T-1}$ 中的信息来预测 $\boldsymbol{z}_T$。我们通过最小化重建误差来训练模型。该损失的最小值为

$$\mathcal{L}_1^* = \min_g \mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{T-1})}[||\boldsymbol{z}_T - g(\boldsymbol{z}_{T-1})||_2^2] = \mathbb{E}_{p(\boldsymbol{z}_{T-1})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1})}(\boldsymbol{z}_T)], \tag{5.7}$$

该值在非参数最优 $g^*(\boldsymbol{z}_{T-1}) = \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1})}[\boldsymbol{z}_T]$处取得，其中 $\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1})}(\boldsymbol{z}_T)$ $= \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1})}\{(\boldsymbol{z}_T - \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1})}[\boldsymbol{z}_T])^\top(\boldsymbol{z}_T - \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1})}[\boldsymbol{z}_T])\}$。现在考虑另一个模型 h，它通过再次最小化重建误差，利用 $\boldsymbol{z}_{T-1}$ 和 $\boldsymbol{z}_{T-2}$ 来预测 $\boldsymbol{z}_T$。这次的最小重建误差为

$$\mathcal{L}_2^* = \min_h \mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{T-1}, \boldsymbol{z}_{T-2})}[||\boldsymbol{z}_T - h(\boldsymbol{z}_{T-1}, \boldsymbol{z}_{T-2})||_2^2] = \mathbb{E}_{p(\boldsymbol{z}_{T-1}, \boldsymbol{z}_{T-2})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1}, \boldsymbol{z}_{T-2})}(\boldsymbol{z}_T)], \tag{5.8}$$

[2]与扩散模型中 $\boldsymbol{z}_0$ 表示原始图像不同，从现在起我们将用 $\boldsymbol{z}_T$ 表示原始图像，因为它是伪视频的末帧。

该值在非参数最优 $h^*(\boldsymbol{z}_{T-1}, \boldsymbol{z}_{T-2}) = \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1},\boldsymbol{z}_{T-2})}[\boldsymbol{z}_T]$处取得。使用更多伪帧来生成目标图像的好处在于，最小重建误差绝不会因使用更多伪帧而增大，这是因为根据全方差公式，

$$\mathcal{L}_2^* - \mathcal{L}_1^* = -\mathbb{E}_{p(\boldsymbol{z}_{T-1})}\{\mathrm{Var}_{p(\boldsymbol{z}_{T-2}|\boldsymbol{z}_{T-1})}(\mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1},\boldsymbol{z}_{T-2})}[\boldsymbol{z}_T])\} \leq 0. \tag{5.9}$$

此外，通过一阶马尔可夫链创建伪视频的非最优性变得清晰：一阶马尔可夫数据增强意味着 $p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1}) = p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1}, \boldsymbol{z}_{T-2})$，从而 $\mathcal{L}_2^* = \mathcal{L}_1^*$。更具体地说，为了在式 5.9 中得到严格不等式，我们需要避免 $p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1}) = p(\boldsymbol{z}_T|\boldsymbol{z}_{T-1}, \boldsymbol{z}_{T-2})$，这等价于避免使用一阶马尔可夫链 $\boldsymbol{z}_T \to \boldsymbol{z}_{T-1} \to \boldsymbol{z}_{T-2}$ 或$\boldsymbol{z}_T \leftarrow \boldsymbol{z}_{T-1} \to \boldsymbol{z}_{T-2}$。这一分析对我们设计更好的伪视频具有启发意义，例如通过高阶马尔可夫链进行数据增强。我们将上述非正式推理形式化为定理5.4.1 ，并在附录 D.1 中给出正式证明。

定理 5.4.1. 考虑两个根据若干先前帧预测最后一帧 $\boldsymbol{z}_T$的视频生成模型。假设它们采取如下形式：$\hat{\boldsymbol{z}}_T^{(g)} = g(\boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \cdots, \boldsymbol{z}_{s_k})$和 $\hat{\boldsymbol{z}}_T^{(h)} = h(\boldsymbol{z}_{s_1}, \boldsymbol{z}_{s_2}, \cdots \boldsymbol{z}_{s_l})$，分别为，其中 $T > s_1 > \cdots > s_k > \cdots > s_l$。于是，我们有

$$\min_{\hat{\boldsymbol{z}}_T^{(h)}} \mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_l})}[||\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(h)}||_2^2] \leq \min_{\hat{\boldsymbol{z}}_T^{(g)}} \mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[||\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(g)}||_2^2], \tag{5.10}$$

等式在 $\boldsymbol{z}_T|\boldsymbol{z}_{s_1}, \cdots \boldsymbol{z}_{s_k} \overset{d}{=} \boldsymbol{z}_T|\boldsymbol{z}_{s_1}, \cdots \boldsymbol{z}_{s_l}$ 时取等。

注 5.4.2. 上述最小重建误差是在非参数最优解下得到的。在实践中，这对应于如下假设：我们的神经网络 g_θ 和 h_ϕ 足够灵活，能够准确逼近非参数最优解，从而使定理成立。此外，该分析基于$\{\boldsymbol{z}_{s_i}\}_{i=1}^l$ 从真实分布中抽取的假设，而在实践中它们还需要与其关联的先前帧一起生成，这意味着当生成的$\{\hat{\boldsymbol{z}}_{s_i}\}_{i=1}^l$ 远离其真实分布时，该定理将不成立。尽管如此，该分析为使用更长历史上下文进行条件生成的益处，以及使用更具表达力的伪视频（而非标准扩散模型中通过一阶马尔可夫转移创建的伪视频）可能带来的性能提升提供了直观认识，我们将在下一节通过实验对此进行实证验证。

5.4.2 实验

我们考虑利用先前已生成帧所提供的信息来生成伪视频的每一帧。具体而言，我们使用视频扩散模型（Harvey等人，2022年），该模型通过在大小为 C的上下文窗口中，对最近的先前帧进行条件化并以自回归方式预测帧来训练。换言之，条件分布 $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-C:t-1})$ 使用由神经网络参数化、并将 $\boldsymbol{z}_{t-C:t-1}$ 作为额外输入的扩散模型进行估计。我们比较了在分别由标准一阶马尔可夫变换和高阶马尔可夫变换创建的伪视频上训练的视频扩散模型的性能，以实证验证我们在第5.4.1节中的论点。下文描述实验设置，详细超参数见附录D.2。

数据集。 我们使用来自 CIFAR10（32×32）和 CelebA（64×64）的图像创建4帧和8帧伪视频。我们使用高斯噪声作为数据增强，并考虑两种策略：

- **一阶马尔可夫**。 我们递归添加高斯噪声3次或7次，以线性调度（Ho 等人，2020年）创建一阶马尔可夫伪视频，其中β的范围从0.0001到0.05[3]：$\boldsymbol{z}_{T-t} = \sqrt{1-\beta_t}\boldsymbol{z}_{T-t+1} + \sqrt{\beta_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$。
- **高阶马尔可夫**。 在使用相同的噪声调度创建 $\boldsymbol{z}_{T-t}$时，我们不是向 $\boldsymbol{z}_{T-t+1}$添加高斯噪声，而是采用一种简单策略，通过对$\{\boldsymbol{z}_{T-t+s}\}_{s=1}^{t}$的均值添加高斯噪声来创建高阶马尔可夫伪视频：$\boldsymbol{z}_{T-t} = \sqrt{1-\beta_t}[\frac{1}{t}\sum_{s=1}^{t}\boldsymbol{z}_{T-t+s}] + \sqrt{\beta_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$。

我们在图5.6中绘制了使用上述两种策略创建的伪视频示例。我们再次使用1帧来表示图像生成模型对应方法——改进型 DDPM（Nichol 和 Dhariwal，2021年）——在原始目标图像上训练所得的结果。我们还将模糊处理作为数据增强加以考虑，但其性能不如使用高斯噪声时的性能（见下文**结果**段落）。

网络架构。 我们采用了与Harvey 等人(2022年类似的 UNet 架构，所有模型的每个下采样层和上采样层都包含 2 个残差块，基础通道大小均为 128。注意，Harvey 等人(2022年) 的模型基于与单帧图像扩散模型（Nichol 和 Dhariwal，2021年）相同的架构构建，因此单帧图像扩散模型也采用相同的超参数。在生成过程中，我们使用Harvey提出的“Autoreg”采样方案

[3]在超参数选择阶段，我们试验了从{0.02、0.05、0.1}中选取的最大 β，其中0.05在一阶马尔可夫数据增强下取得了最佳性能。



(a) 一阶



(b) 高阶

图 5.6：通过使用一阶马尔可夫链（上）和高阶马尔可夫链（下）向 CIFAR10 图像添加高斯噪声而构造的伪视频示例。

表5.3：在由 CIFAR10 和 CelebA 图像构造的伪视频上训练的视频扩散模型生成图像的最后一帧FID（采用一阶马尔可夫或高阶马尔可夫高斯噪声数据增强）。单帧结果来自在原始 CIFAR10 和 CelebA 图像上训练、采用相同 UNet 架构的图像扩散模型。

	CIFAR10			CelebA		
	1帧	4帧	8帧	1帧	4帧	8帧
一阶马尔可夫	12.90	17.30	15.90	7.76	13.61	12.64
高阶马尔可夫	12.90	**12.58**	12.80	7.76	**6.88**	7.55

等人 (2022年)，从而每一帧 $\boldsymbol{z}_t$ 通过对上下文窗口中最近生成的帧进行条件化而生成，$\{\boldsymbol{z}_{t-c}\}_{c=1}^{C}$。上下文窗口的大小$C$ （即时间滞后）对于4帧模型和8帧模型分别为2和4。每次生成新的一帧时，我们考虑1,000个扩散步数。由于4帧模型和8帧模型在开始时分别联合生成前2帧和前4帧（初始上下文窗口），它们分别总共使用3,000和5,000个扩散步数来生成完整的伪视频。我们还考虑在训练单帧图像扩散模型时将扩散步数从1,000增加到4,000，并将其与总共具有3,000个扩散步数的4帧视频扩散模型进行比较，以确保视频扩散模型的性能提升并非仅仅因为我们总体上有更多的扩散步数。

结果。 我们再次计算 FID（基于 1 万个样本）来评估模型。表 5.3 分别展示了视频扩散模型为 CIFAR10 和 CelebA 图像生成的伪视频的最后一帧 FID。1帧的结果对应于其图像对应模型（即改进型 DDPM）的性能。尽管视频



(a) CIFAR10 (b) CelebA

图 5.7：分别由在 CIFAR10 和 CelebA 的 4 帧高阶马尔可夫伪视频上训练的视频扩散模型生成的图像。

在一阶马尔可夫伪视频上训练的视频扩散模型并未优于单帧图像扩散模型，但在高阶马尔可夫伪视频上训练的 4帧和 8帧视频扩散模型在两个数据集上都取得了更好的结果。这从经验上证明了如第 5.4.1 节所示，在将目标图像中的信息传递至伪帧方面，一阶马尔可夫链并非最优；同时也支持了我们的提议，即使用表达能力更强的伪视频，而非由一阶马尔可夫链生成的伪视频。请注意，4帧模型优于 8帧模型，这可能是因为较长伪视频的真实分布性质更复杂，因此可能需要表达能力更强的架构才能达到最优结果（见备注 5.4.2）。不过，为了进行公平的实践比较，我们在所有模型中使用相同的 UNet架构。再次强调，这种反向趋势在实践中不应是严重问题，因为实践者可能更倾向于用尽可能少的伪帧来改进生成，以降低额外的计算成本。我们在图 5.7 中展示了在 CIFAR10 和 CelebA 上训练的 4帧模型生成的一些图像。

表 5.4 比较了4帧模型与扩散步数为4,000的单帧模型。尽管扩散步数总量更多的单帧模型在CIFAR10上的性能有所提升，并且优于4帧模型，但它在CelebA上的表现却不如4帧模型。此外，在CelebA上，它的表现甚至不如扩散步数仅为1,000的基线单帧模型，而4帧视频扩散模型在两个数据集上的性能都持续提升。这表明，单纯增加图像扩散模型的扩散步数可能并不总是有效。

表 5.4：在由 CIFAR10 和 CelebA 图像构造的伪视频上，使用高阶马尔可夫高斯噪声数据增强进行训练的视频扩散模型所生成图像的最后一帧 FID。1帧结果来自在原始 CIFAR10 和 CelebA 图像上训练的图像扩散模型。此处，1帧模型使用 4,000 个扩散步数，而 4帧模型总共使用 3,000 个扩散步数。

	1帧（1k 步）	1帧（4k 步）	4帧（总计 3k 步）
CIFAR10	12.90	**11.95**	12.58
CelebA	7.76	7.87	**6.88**

除了使用高斯噪声，我们还尝试像第5.3节那样使用高斯模糊来创建伪视频。然而，我们在CIFAT10上进行的高斯模糊实验结果不如添加高斯噪声（见表 5.5），因此决定不在后续实验中考虑这种方法。这表明在实践中，表现良好的数据增强策略可能因不同类别的视频生成模型而异。

表 5.5：在采用高阶马尔可夫数据增强（高斯噪声或高斯模糊）从 CIFAR10 图像构造的伪视频上训练的视频扩散模型所生成图像的最后一帧FID。

	4帧	8帧
高斯噪声	**12.58**	12.80
高斯模糊	15.33	22.63

5.5 相关工作

5.5.1 序列生成模型

层次变分自编码器（HVAEs）（Sønderby 等人，2016年；Maaløe 等人，2019年；Vahdat 和 Kautz，2020年；Child，2021年；Xiao 和 Bamler，2023年）是一类通过堆叠标准变分自编码器（Kingma 和 Welling，2014年）构建的序列生成模型。尽管分层变分自编码器代表了一类表达能力丰富的生成模型，但由于优化困难，它们在实践中难以训练，如第 5.2 节所述。扩散模型（Sohl-Dickstein 等人，2015年；Ho 等人，2020年；Song 等人，2021b；Kingma 等人，2021年；Nichol 和 Dhariwal，2021年；Song 等人，2021a；Rissanen 等人，2023年；Bansal 等人，2023年；Hoogeboom 和 Salimans，2023年）可被视为分层变分自编码器的一种特例，其中编码器是固定的、预定义的高斯卷积核。具体而言，它们本质上是对一系列噪声图像进行回归，这些图像创建自

目标图像，并借助自监督，如第 5.2 节所述。尽管与分层变分自编码器存在相似度，扩散模型以及潜在扩散模型（Rombach 等人，2022年）在另一个潜变量模型（例如，VQVAE（Van Den Oord 等人，2017年））的较低维度潜在空间中应用扩散模型，已取得最先进性能，部分得益于噪声污染图像所提供的额外自监督信号。流匹配（Lipman 等人，2023年；Liu 等人，2022年；Albergo 等人，2023年；Gat 等人，2024年；Wang 等人，2024年）是另一种最先进的序列生成建模技术，它通过回归一系列诱导概率路径的向量场来训练连续归一化流（Chen 等人，2018年），该概率路径以直接自监督的方式连接数据分布与先验分布。已有研究表明，流匹配能够学习比扩散模型更直的轨迹，从而在生成时需要更少的离散化步数。此外，流匹配使我们能够放宽对先验分布的高斯假设，从而实现任意分布之间的耦合（Albergo 等人，2023年）。相比之下，我们提出的框架引入了一类新方法，利用视频生成模型和带有自监督帧的伪视频来改进任意给定的图像生成模型。

5.5.2 自监督学习

自监督学习（Liu 等人，2021年；Shwartz Ziv 和 LeCun，2024年）通过为无标签数据手工构造伪标签，将无监督学习问题转化为监督学习问题。自监督学习有两种常见方法。1）对比学习（Chen 等人，2020b年；Tian 等人，2020年；Wu 等人，2020年），预测两个输入是否为同一原始数据的不同数据增强。2）掩码学习（Devlin 等人，2019年；He 等人，2022年；Fang 等人，2023年）在给定未掩码部分的情况下，预测输入中被随机掩码的部分。虽然我们将视频模型拟合到通过对原始图像进行数据增强而创建的伪视频序列这一方法并不属于这两类，但它本质上是一种新的自监督学习形式，因为这些伪视频序列可被视为供我们的模型预测的手工构造伪标签，从而为模型提供额外信息（例如，原始图像的不同保真度）。

5.6 结论与讨论

总结。 我们通过比较标准分层变分自编码器和扩散模型得出了关键见解：扩散模型中噪声污染的伪帧所提供的关于中间状态的额外自监督信息，可能是其成功的原因。基于这一见解，我们提出利用对目标图像应用数据增强所构建的伪视频中的自监督信息

来提升图像生成模型的性能。具体做法是，将图像生成模型扩展为对应的视频生成模型，并在伪视频上训练视频生成模型。实验结果表明，对于两种流行的图像生成模型 VQVAE 和改进型 DDPM，在仅有少量帧的伪视频上训练其对应的视频生成模型，能够提升图像生成性能；这一结果以实证方式验证了伪视频中额外自监督信息的益处。

讨论与未来工作。我们提出的框架为扩展任意给定的生成模型提供了一种替代方案：我们没有通过叠加更多层来增大生成模型，而是展示了这样一种可能性：将一个在图像上训练的图像生成模型转化为相应的视频生成模型，并在伪视频上进行训练，从而提高生成质量。由于许多视频生成模型都建立在图像生成模型之上，这种做法通常很直接。另一方面，这也带来了如何设计信息丰富的伪视频这一挑战。在自回归视频生成框架中，我们从理论上揭示了一阶马尔可夫伪视频可能存在的问题，并提出改用高阶马尔可夫伪视频来解决这一问题。然而，在如此广阔的设计空间中，何种伪视频才是最优选择，通常并不明确；我们将此留作未来的研究问题。另一个有趣的未来方向是探索同一原理能否应用于其他数据模态。对于图像而言，借助数据增强很容易获得自监督信号，但对于文本或分子等其他数据模态，是否存在恰当的方法来注入自监督信息，目前仍不清楚。

第 6 章

结论与未来工作

本学位论文致力于弥合深度序列模型与概率方法之间的差距，以开发兼具两个领域优势的新型机器学习模型。我们努力利用深度序列模型中的归纳偏置与现有概率方法之间的天然联系来指导模型设计，使所提出的方法能够与深度序列模型领域已有的进展相兼容。第 3 章和第 4 章分别利用流行的深度序列模型架构与高斯过程之间的天然联系，构建分别适用于 Transformer 架构和在线学习问题的高斯过程模型。第 5章受到扩散模型与 HVAE 对比的启发，提出通过联合建模原始图像序列及其利用自监督信息构造的伪视频，来改进图像生成模型。下文将在第 6.1节详细回顾各章的贡献。在第6.2节，我们将批判性地概述所提方法的局限，以及弥补这些局限的潜在未来工作。

6.1 学位论文总结

在第 3 章中，我们发现了核注意力与稀疏变分高斯过程（SVGP）后验均值之间的等价关系：核注意力模块中的查询、键和值，分别等价于 SVGP 中与后验均值相关联的查询输入位置、诱导点位置和变分参数。基于这一关键观察，我们提出了稀疏高斯过程注意力（SGPA），它为注意力输出设置高斯过程先验，并利用对应 SVGP 的后验协方差（基于额外的变分协方差参数构造）来量化注意力输出的不确定性。我们将多个 SGPA 层堆叠起来，构建概率 Transformer；这类模型本质上是深度高斯过程，并通过变分推断

训练基于 SGPA 的 Transformer。SGPA 专为 Transformer 架构设计，直接在注意力输出空间中量化不确定性，而不像权重空间贝叶斯神经网络方法那样通过权重的后验分布来量化。对于基于自注意力的 Transformer，标准 SGPA 需要通过摊销获得依赖于输入的变分协方差，这与依赖于输入的键和值类似。为此，我们进一步提出了解耦 SGPA，它借鉴了解耦 SVGP 中的技术来降低计算复杂度；其变分协方差完全基于另一组在所有输入序列间共享的全局键构造。我们在一系列预测任务上对 SGPA 进行了实证评估，结果表明，它能有效校准 Transformer 模型中的不确定性，同时保持具有竞争力的高准确率。

在第4章中，我们发现，在 HiPPO（高阶多项式投影算子）框架中以高斯过程先验替代确定性信号后，可以自然地将 HiPPO 时变正交投影解释为域间稀疏变分高斯过程的诱导变量，其基函数为时变正交多项式。这些域间诱导变量能够自适应地压缩高斯过程历史，同时保留长期记忆。基于此，我们开发了在线 HiPPO-SVGP（OHSVGP），利用 HiPPO 保留长程记忆的能力来处理在线学习任务。OHSVGP 通过高效的 HiPPO-ODE 递推在线更新先验协方差矩阵，绕过了标准在线 SVGP 中基于优化、繁琐的在线诱导位置更新，从而进一步提高了 OHSVGP 的计算效率。我们通过一系列在线学习和持续学习任务对 OHSVHP 进行了实证评估，并证明 OHSVGP 的表现优于现有的在线稀疏高斯过程方法，尤其是在需要长期记忆的场景中。高效的流式处理能力和对历史信息的保留，使 OHSVGP 非常适合应用于同时要求速度和准确率的现实世界在线学习场景。

在第5章，我们将关注点从预测任务转向生成建模。具体而言，我们探讨了序列或序列生成建模的益处。我们首先比较了两类序列生成模型，即扩散模型与分层变分自编码器。我们的关键观测是，扩散模型可被视为分层变分自编码器，但它们以通过破坏原始目标图像所创建的伪视频的形式，为中间变量序列施加直接监督信号，而标准的分层变分自编码器则使这些中间状态完全灵活，这或许是扩散模型优于分层变分自编码器的原因之一。受这一差异启发，我们探索了通过联合建模原始图像及其包含自监督信息的对应伪视频的分布，来改进其他类型图像生成模型的可能性。

具体而言，我们将两种流行的图像生成模型框架——VQVAE 和改进型 DDPM——扩展为对应的视频生成模型，并在伪视频上对其进行训练。我们通过实验表明，该流程仅用几帧伪视频即可提升图像生成质量。此外，我们从理论上分析了在自回归视频生成框架中，伪视频上何种概率结构可能是可取的。这一分析使我们能够识别某些伪视频的潜在问题，包括呈一阶马尔可夫链形式的伪视频，并提出了一种简单有效的方法，通过构建高阶马尔可夫伪视频来规避这些潜在问题。

6.2 局限性与未来研究方向

我们展示了这样一种方法的潜力：在构建针对深度序列模型的概率方法时，充分考虑深度序列模型中的归纳偏置。我们相信，这一愿景能够启发更多未来工作，将深度序列模型与概率建模结合起来，从而实现相互促进、共同改进。我们列出了一些未来研究方向，旨在克服本学位论文所提方法的局限性，进一步推动改进。

将 SGPA 扩展至基于解码器的 Transformer。在第3章中，我们仅考虑将 SGPA 应用于基于编码器的 Transformer，其中序列上下文是固定的。然而，许多序列建模任务需要基于基于解码器的 Transformer 进行自回归预测（Devlin 等人，2019年；OpenAI 等人，2024b）。例如，在自然语言应用中，基于解码器的 Transformer 常用于根据由先前生成的词元构成的现有上下文预测下一个词元。由于上下文会持续更新，第3章中的静态高斯过程先验便不适合表示自适应上下文之外不断变化的未知信息所带来的动态不确定性。序贯贝叶斯推断会不断将当前先验更新为上一个时间步的（近似）后验，因此更适合对这种自适应上下文知识进行建模。为通过变分推断实现这一点（Nguyen 等人，2018年），我们可以用以下在线证据下界替换面向基于编码器的 SGPA 的静态证据下界训练目标：

$$\begin{aligned}\mathcal{L}_{ELBO}^{(t)} = &\mathbb{E}_{q_t(\boldsymbol{F}^{L,t}|\boldsymbol{F}^{0,t},\{\boldsymbol{k}^{l,h,t}\}_{l=1,h=1}^{L,H})}[\log p(\boldsymbol{Y}^t|\boldsymbol{F}^{L,t})] \\ &-\mathrm{KL}(q_t(\boldsymbol{F}^{L,t}|\boldsymbol{F}^{0,t},\{\boldsymbol{k}^{l,h,t}\}_{l=1,h=1}^{L,H})||q_{t-1}(\boldsymbol{F}^{L,t-1}|\boldsymbol{F}^{0,t-1},\{\boldsymbol{k}^{l,h,t-1}\}_{l=1,h=1}^{L,H})),\end{aligned} \tag{6.1}$$

其中， q_{t-1} 在前一个时间步获得，并成为当前步推断中 KL 正则项所用的先验。注意， q_t 和 q_{t-1} 二者均为

深度稀疏高斯过程，但键的集合不同，因此，得益于先验条件匹配条件，上述在线证据下界中的 KL 正则项可再次简化为完全基于诱导键和值计算的 KL 项，这与 OHSVGP 的在线证据下界（式 4.6）类似。对于自回归预测任务，时间步 t（$\boldsymbol{k}^{l,h,t}$ 和 $\boldsymbol{v}^{l,h,t}$）的诱导键和值，是在先前的诱导键和值（$\boldsymbol{k}^{l,h,t-1}$ 和 $\boldsymbol{v}^{l,h,t-1}$）的基础上，增添一个根据最近生成的词元计算得到的键和值而获得的（这一过程称为键值缓存）。

更高效的 SGPA。与标准Transformer相比，SGPA由于需要额外计算后验协方差，计算成本更高。在第3章中，我们已经使用解耦稀疏变分高斯过程技术，仅基于所有序列数据点共享的固定数量全局诱导键来构建后验协方差。然而，要将 SGPA 扩展到现代应用中部署的大型 Transformer（例如大语言模型），进一步降低计算成本必不可少。例如，可使用随机傅里叶特征（RFF；（Rahimi 和 Recht，2007年））近似核，以加速核矩阵的计算（Reid 等人，2023年）。此外，还可以调整 SGPA 的架构，以支持更优的推断方案。基于某些特殊类型的核，存在计算 SGPA 后验分布的等效但成本更低的方法。例如，无需计算完整协方差，基于马尔可夫核的高斯过程后验采样可以通过线性随机微分方程（SDE）框架中的平滑来实现（Hartikainen 和 S¨arkk¨a，2010）。

HiPPO-SGPA。 对于基于解码器的 Transformer，生成新词元时键值对的数量会不断增加，因此很难构造合适的后验协方差，在建模不断增长的上下文中的自适应知识的灵活性与计算效率之间取得平衡。标准 SGPA 基于数量不断增长的（T）摊销诱导键构造变分协方差，因此其规模也会持续增长，导致计算开销很大。解耦 SGPA 则基于固定数量的（M）静态诱导键构造变分协方差，这些键在所有序列间全局共享；但当序列数量非常庞大时（例如，用于训练大语言模型的大型语料库），这种做法可能缺乏足够的灵活性，无法概括整个序列数据集中的全局知识。理想情况下，我们希望为每个序列维护固定数量的自适应诱导键值对，并动态更新它们，再基于这些键值对构造固定规模的自适应变分协方差。第 4章介绍的 HiPPO-SVGP 提供了一种可行方案，因为 HiPPO 会维持固定规模的自适应记忆状态，以捕捉动态历史。给定

一组查询后，我们可以使用第 4.4 节介绍的 HiPPO递推，得到跨域注意力矩阵 $\boldsymbol{K}_{fqu}$；在自回归预测过程中收到新的查询时，跨域注意力矩阵将通过另一个递推步骤进行更新，以纳入新查询的信息（式 4.18）。SGPA 计算所需的最后几个组成部分，是基于这些固定规模的跨域诱导键构造的固定规模自适应变分值和协方差。标准 Transformer 使用施加于注意力模块序列输入上的投影矩阵来获取这些量，但该方法的规模会随输入自适应变化 T。一种可行方案是改为对序列输入再应用一个循环神经网络，动态输出所需的固定规模变分参数。

基于 HiPPO-SVGP 构建概率状态空间模型。HiPPO-SVGP 也可以扩展用于构建概率状态空间模型。标准状态空间模型层利用结构化递推获取记忆状态，以概括序列中的信息，然后将记忆状态投影以获得输出（Gu 等人，2022年；Gu 和 Dao，2023年）。在 HiPPO-SVGP 中，记忆状态（或系数）的概念等同于域间诱导点，并非确定性变量，而是服从变分高斯分布 $q(\boldsymbol{u})$。基于此，我们可以构建概率状态空间模型层，从而实现对状态空间模型的不确定性量化。不过，我们仍需找到合理的方法，将这些概率状态空间模型层堆叠起来，以构建更灵活的深度模型。此外，尽管递推参数的结构是根据固定的 HiPPO递推矩阵设计的，但在最先进的深度状态空间模型中，这些参数是可训练的（Gu 等人，2022年；Gu 和 Dao，2023 年）。因此，我们也可以考虑调整 HiPPO-SVGP 中的递推参数。在这种情况下，训练得到的递推参数将对应于某些隐式函数基和测度，而我们并不知道它们的解析形式。

更好的伪视频设计。 正如我们在第 5 章所展示的，在伪视频生成框架中，更好的伪视频设计可能会提升图像生成质量。一般而言，由于伪视频框架将图像生成模型作为其特例（即 $T = 1$），直观来看，应该存在一组超参数配置（例如数据增强类型、架构等），使其性能优于图像（1帧）生成。然而，正如我们在第 5.6 节所提到的，伪视频框架大大拓展了超参数空间，尤其是用于创建伪视频的数据增强设计空间极其庞大。对于不同类型的模型和架构，最优的数据增强通常并不相同（例如，实证结果表明，高斯模糊数据增强对 C-ViViT 的效果更好，但在第 5 章中，其效果不如用于视频扩散的高斯噪声）。尽管我们从理论上说明了一些直觉，即高阶

阶数马尔可夫数据增强更适用于自回归视频生成模型，但如何找到最优的数据增强，或为其他类型的模型确定更合适的数据增强实用建议，仍是一个开放的研究问题。尽管如此，我们相信，我们的分析和实证实验已证明这一框架具有可行性，并且潜力巨大，有望提升生成性能。

参考文献

Albergo, M. S., Boffi, N. M., 和 Vanden-Eijnden, E.（2023年）。随机插值：统一流与扩散过程的框架。*arXiv* 预印本 *arXiv:2303.08797*。Arbel, J., Pitas, K., Vladimirova, M., 和 Fortuin, V.（2023年）。贝叶斯神经网络入门：综述与争论。*arXiv* 预印本 *arXiv:2309.16314*。Ashman, M., So, J., Tebbutt, W., Fortuin, V., Pearce, M., 和 Turner, R. E.（2020年）。稀疏高斯过程变分自编码器。*arXiv* 预印本 *arXiv:2010.10177*。Ba, J. L., Kiros, J. R., 和 Hinton, G. E.（2016年）。层归一化。载于*arXiv* 预印本 *arXiv:1607.06450*。

Bansal, A., Borgnia, E., Chu, H.-M., Li, J., Kazemi, H., Huang, F., Goldblum, M., Geiping, J., and Goldstein, T.（2023年）。冷扩散：无需噪声即可反演任意图像变换。神经信息处理系统进展，36。

Bayes, T.（1763）。LII. 关于解决概率论中一个问题的论文。已故牧师 Bayes 先生所著，由 FRS 通过 Price 先生以信件形式转交给 AMFR S. 的 John Canton。《伦敦皇家学会哲学汇刊》，53：370–418。

Beal, M. J. (2003). 用于近似贝叶斯推断的变分算法. 伦敦大学博士学位论文.
Benita, R., Elad, M., and Keshet, J. (2023). Diffar：用于原始语音波形生成的去噪扩散自回归模型. 载于*arXiv*预印本 *arXiv:2310.01381*. Blair, M., Tho mpson, J., Henrey, A., and Chen, B. (2013). Skill- Craft1 主表数据集. UCI机器学习资料库. 数字对象唯一标识符：https://doi.org/10.24432/C5161N.
Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). 神经网络中的权重不确定性. 载于国际机器学习会议.

Bradshaw, J., Matthews, A. G. d. G., and Ghahramani, Z. (2017). 高斯过程混合深度网络中的对抗样本、不确定性、迁移测试与鲁棒性. *arXiv*预印本 *arXiv:1707.02476*. Brooks, S., Gelman, A., Jones, G., and Meng, X. L. (2011). 马尔可夫链蒙特卡罗手册. CRC出版社. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., S astry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). 语言模型是少样本学习者. 神经信息处理系统进展, 33:1877–1901. Bui, T. D., Nguyen, C. V., and Turner, R. E. (2017). 流式稀疏高斯过程近似. 载于神经信息处理系统进展. Bui, T. D. and Turner, R. E. (2014). 树结构高斯过程近似. 载于神经信息处理系统进展.

Burda, Y., Grosse, R., 和 Salakhutdinov, R.（2015年）。重要性加权自编码器。*arXiv* 预印本 *arXiv:1509.00519*。

Burt, D. R., Rasmussen, C. E., 和 van der Wilk, M.（2019年）。稀疏变分高斯过程回归的收敛速率。载于国际机器学习会议（*ICML*）。

Campbell, A., Harvey, W., Weilbach, C., De Bortoli, V., Rainforth, T., 和 Doucet, A.（2023年）。通过跳跃扩散模型进行跨维度生成建模。载于神经信息处理系统进展。

Casale, F. P., Dalca, A., Saglietti, L., Listgarten, J., 和 Fusi, N.（2018年）。高斯过程先验变分自编码器。神经信息处理系统进展，31。

Cesa-Bianchi, N. 和 Lugosi, G.（2006年）。预测、学习与博弈。剑桥大学出版社。

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. (2022年). MaskGit：掩码生成图像变换器。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 11315–11325 页。

Chang, P. E., Verma, P., John, S., Solin, A., and Khan, M. E. (2023年). 基于记忆的双重高斯过程用于序贯学习。载于国际机器学习会议。Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. (2020年a). 基于像素的生成式预训练。载于国际机器学习会议，第 1691–1703 页。PMLR。Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018年). 神经常微分方程。神经信息处理系统进展，31。

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G.（2020b）。用于视觉表征对比学习的简单框架。载于国际机器学习会议，第1597–1607页。PMLR。

Chen, W., Chen, W., Rastrelli, L., and Li, Y.（2025a）。你的图像其实是伪视频的末帧。载于机器学习中的深度生成模型：理论、原理与效能（*DeLTa*）*ICLR*研讨会。

Chen, W., Kiyohara, N., Zhu, H. B. H., Curran-Sebastian, J., Bhatt, S., and Li, Y.（2025b）。在线域间高斯过程的循环记忆。载于神经信息处理系统进展（*NeurIPS*）。

Chen, W., Klochkov, Y., and Liu, Y.（2024年）。事后偏差评分对于公平分类是最优的。载于国际学习表征会议（*ICLR*）。

Chen, W., Li, B., Zhang, R. 和 Li, Y.（2025c）。深度学习中的贝叶斯计算。载于*arXiv* 预印本 *arXiv:2502.18300*。

Chen, W. and Li, Y.（2023）。通过稀疏高斯过程校准变换器。发表于国际学习表征会议（*ICLR*）。Cheng, C.-A. and Boots, B.（2017）。线性复杂度高斯过程模型的变分推断。发表于 神经信息处理系统进展。Child, R.（2021）。超深变分自编码器具有泛化能力，并且在图像任务上的表现优于自回归模型。发表于 国际学习表征会议。Cinquin, T., Immer, A., Horn, M., and Fortuin, V.（2021）。贝叶斯 Transformer 的先验与推断中的病态问题。发表于 *NeurIPS 2021*“我简直不敢相信它没有更好”研讨会。

Coker, B., Bruinsma, W. P., Burt, D. R., Pan, W., and Doshi-Velez, F.（2022）。宽平均场变分贝叶斯神经网络忽略数据。发表于 人工智能与统计国际会议。Csat´o, L. and Opper, M.（2002年）。稀疏在线高斯过程。神经计算，14(3):641–668。Damianou, A. C. and Lawrence, N. D.（2013年）。深度高斯过程。发表于人工智能与统计国际会议。Dao, T. and Gu, A.（2024年）。Transformer 即状态空间模型：通过结构化状态空间对偶性实现的广义模型与高效算法。发表于国际机器学习会议（*ICML*）。Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.（2019年）。BERT：用于语言理解的深度双向 Transformer 预训练。发表于 计算语言学协会北美分会年会。Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., and Gelly, S.（2021年）。一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer。发表于 国际学习表征会议。Dutordoir, V., Durrande, N., and Hensman, J.（2020年）。具有球谐特征的稀疏高斯过程。发表于国际机器学习会议（*ICML*）。Duvenaud, D. K., Nickisch, H., and Rasmussen, C.（2011年）。加性高斯过程。发表于神经信息处理系统进展。Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X.（2020年）。图神经网络基准测试。*arXiv* 预印本 *arXiv:2003.00982*。Fan, X. F., Zhang, S., Chen, B., and Zhou, M.（2020年）。贝叶斯注意力模块。发表于神经信息处理系统进展。Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., and Cao, Y.（2023）。EVA：探索大规模掩码视觉表征学习的极限。发表于*IEEE/CVF*计算机视觉与模式识别会议论文集，第19358–19369页。Foong, A. Y. K., Burt, D., Li, Y., and Turner, R.（2020年）。论贝叶斯神经网络中近似推断的表达能力。发表于 神经信息处理系统进展。

Fortuin, V., Baranchuk, D., R¨atsch, G., and Mandt, S. (2020). GP-VAE：深度概率时间序列插补。载于人工智能与统计国际会议，第 1651–1661 页。PMLR。Gal, Y. (2016). 深度学习中的不确定性。博士学位论文，剑桥大学。Gal, Y. and Ghahramani, Z. (2016). 将 Dropout 作为贝叶斯近似：在深度学习中表示模型不确定性。载于国际机器学习会议。Gal, Y. and Turner, R. E. (2015). 通过表示频率输入的不确定性，改进高斯过程稀疏谱近似。载于国际机器学习会议 *(ICML)*。Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. (2016). 神经网络的领域对抗训练。机器学习研究期刊，17(59)：1–35。Garifolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., Dahlgren, N., and Zue, V. (1993). TIMIT 声学语音连续语料库 LDC93S1。载于费城：语言数据联盟。Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T., Synnaeve, G., Adi, Y., and Lipman, Y. (2024). 离散流匹配。*ar Xiv* 预印本 *arXiv:2407.15595*。Gelfand, A. E. (2000). 吉布斯采样。美国统计学会会刊，95(452)：1300–1304。Gershman, S. J. and Goodman, N. D. (2014). 概率推理中的摊销推断。认知科学学会年会论文集，36。Gneiting, T., Balabdaoui, F., and Raftery, A. E. (2007). 概率预测、校准与尖锐度。皇家统计学会会刊：*B*辑（统计方法学），69(2)：243–268。Graves, A., Jaitly, N., and Mohamed, A.-r. (2013a). 基于深度双向长短期记忆网络的混合语音识别。载于 *2013*年 *IEEE* 自动语音识别与理解研讨会，第 273–278 页。IEEE。Graves, A., Mohamed, A.-r., and Hinton, G. E. (2013b). 基于深度循环神经网络的语音识别。载于*2013*年 *IEEE* 声学、语音与信号处理国际会议。

Gu, A. and Dao, T. (2023). Mamba：使用选择性状态空间进行线性时间序列建模。*arXiv* 预印本 *arXiv:2312.00752*。Gu, A., Dao, T., Ermon, S., Rudra, A., and R´e, C. (2020). HiPPO：采用最优多项式投影的循环记忆。载于神经信息处理系统进展。Gu, A., Goel, K., and R´e, C. (2022). 使用结构化状态空间高效建模长序列。载于国际学习表征会议。Gu, A., Johnson, I., Timalsina, A., Rudra, A., and Re, C. (2023). 如何训练你的 HIPPO：采用广义正交基投影的状态空间模型。载于国际学习表征会议。Gulrajani, I., Kumar, K., Ahmed, F., Taiga, A. A., Visin, F., Vazquez, D., and Courville, A. (2017). PixelVAE：用于自然图像的潜变量模型。载于国际学习表征会议。Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). 论现代神经网络的校准。载于国际机器学习会议。

Hartikainen, J. 和 S¨arkk¨a, S.（2010）。时间高斯过程回归模型的卡尔曼滤波与平滑解。载于*IEEE*机器学习信号处理国际研讨会。

Harvey, W., Naderiparizi, S., Masrani, V., Weilbach, C., and Wood, F. (2022). 长视频的灵活扩散建模。载于神经信息处理系统进展。Hawryluk, I., Hoeltgebaum, H., Mishra, S., Miscouridou, X., Schnekenberg, R. P., Whittaker, C., Vollmer, M., Flaxman, S., Bhatt, S., and Mellan, T. A. (2021). 高斯过程临近预报：应用于 COVID-19 死亡率报告。载于人工智能中的不确定性，第 1258–1268 页。PMLR。He, K., Chen, X., Xie, S., Li, Y., Doll´ar, P., and Girshick, R. (2022). 掩码自编码器是可扩展的视觉学习器。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 16000–16009 页。He, K., Zhang, X., Ren, S., and Sun, J. (2015). 用于图像识别的深度残差学习。*arXiv:1512.03385*。Hendrycks, D. and Dietterich, T. (2019). 对神经网络在常见损坏和扰动下的鲁棒性进行基准测试。载于国际学习表征会议。

高斯过程的变分傅里叶特征。机器学习研究期刊，18(151):1–52。Hensman, J., Fusi, N., and Lawrence, N. D. (2013). 面向大数据的高斯过程。载于人工智能不确定性会议。Hensman, J., Matthews, A. G. d. G., and Ghahramani, Z. (2015). 可扩展的变分高斯过程分类。载于人工智能与统计国际会议。Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Hor´anyi, A., Mu˜noz Sabater, J.,Nicolas, J., Peubey, C., Radu, R.,Rozum, I., Schepers,D., Si mmons,A., Soci, C.,Dee,D., andTh´epaut, J. N. (2023). ERA5自1940年至今的单层逐小时数据。哥白尼气候变化服务（C3S）气候数据存储库（CDS）。https://doi.org/10.24381/cds.adbb2d47。Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). 采用双时间尺度更新规则训练的生成对抗网络会收敛到局部纳什均衡。载于神经信息处理系统进展。Ho, J., Jain, A., and Abbeel, P. (2020). 去噪扩散概率模型。神经信息处理系统进展。Hoogeboom, E. and Salimans, T. (2023). 模糊扩散模型。载于国际学习表征会议。Horn, R. A. and Johnson, C. R. (1991). 矩阵分析专题。剑桥大学出版社。Huang, C.-W., Lim, J. H., and Courville, A. C. (2021). 关于基于扩散的生成模型与分数匹配的变分视角。神经信息处理系统进展，34:22863–22876。Jafraste h, B., Villacampa-Calvo, C., and Hernandez-Lobato, D. (2022). 输入依赖型稀疏高斯过程。载于国际机器学习会议。Jayasekera, I. S., Si, J., Valdettaro, F., Chen, W., Faisal, A. A., and Li, Y. (2025). 上下文学习的变分不确定性分解。载于神经信息处理系统进展 *(NeurIPS)*。Jazbec, M., Ashman, M., Fortuin, V., Pearce, M., Mandt, S., and R¨atsch, G. (2021). 可扩展高斯过程变分自编码器。载于人工智能与统计国际会议，第 3511–3519 页。PMLR。

Johnson, J.、Alahi, A. 和 Fei-Fei, L.（2016年）。用于实时风格迁移和超分辨率的感知损失。载于计算机视觉—*ECCV 2016*：第*14*届欧洲会议，阿姆斯特丹，荷兰，*2016*年*10*月*11*日至*14*日，会议录，第二部分 *14*，第694–711页。施普林格。Jordan, M. I.、Ghahramani, Z.、Jaakkola, T. S. 和 Saul, L. K.（1999）。图模型的变分方法引言。机器学习，37(2)：183–233。Jumper, J.、Evans, R.、Pritzel, A.、Green, T.、Figurnov, M.、Ronneberger, O.、ˇ Tunyasuvunakool, K.、Bates, R.、Z´ıdek, A.、Potapenko, A.、Bridgland, A. 等（2021年）。借助 AlphaFold 实现高精度蛋白质结构预测。《自然》，596：583–589。Jung, Y.、Lee, H.、Chen, W.、M¨ollenhoff, T.、Li, Y.、Lee, J. 和 Khan, M. E.（2025a）。持续逻辑回归的紧凑记忆。载于神经信息处理系统进展（*NeurIPS*）。Jung, Y.、Lee, H.、Chen, W.、M¨ollenhoff, T.、Li, Y.、Lee, J. 和 Khan, M. E.（2025b）。基于 k先验的持续学习的紧凑记忆。载于近似贝叶斯推断进展研讨会（*AABI*）。

Kapoor, S.、Karaletsos, T. 和 Bui, T. D.（2021年）。用于持续学习的变分自回归高斯过程。载于国际机器学习会议。

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T.（2020年）。分析并提升 StyleGAN 的图像质量。载于*IEEE/CVF*计算机视觉与模式识别会议，第 8110–8119 页。Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A.（2020年）。变分自编码器与非线性独立成分分析：一个统一框架。载于人工智能与统计国际会议，第 2207–2217 页。PMLR。Kim, J., Nguyen, T. D., Min, S., Cho, S., Lee, M., Lee, H., and Hong, S.（2022年）。纯 Transformer 是强大的图学习器。$arXiv$，abs/2207.02505。Kingma, D., Salimans, T., Poole, B., and Ho, J.（2021年）。变分扩散模型。神经信息处理系统进展，34:21696–21707。Kingma, D. P. and Ba, J.（2015年）。Adam：一种随机优化方法。载于国际学习表征会议。

Kingma, D. P. and Welling, M.（2014年）。自编码变分贝叶斯。载于国际学习表征会议。Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., 等（2017年）。克服神经网络中的灾难性遗忘。美国国家科学院院刊，114(13):3521–3526。Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S.（2017年）。自归一化神经网络。载于神经信息处理系统进展。Kristiadi, A., Hein, M., and Hennig, P.（2020年）。即使只具备些许贝叶斯特性，也能纠正 ReLU 网络的过度自信问题。载于国际机器学习会议。Krizhevsky, A., Nair, V., and Hinton, G.（2009年）。CIFAR-10 和 CIFAR-100 数据集。Krizhevsky, A., Sutskever, I., and Hinton, G. E.（2012年）。使用深度卷积神经网络进行 ImageNet 分类。载于神经信息处理系统进展。Lakshminarayanan, B., Pritzel, A., and Blundell, C.（2017年）。使用深度集成进行简单且可扩展的预测不确定性估计。载于神经信息处理系统进展。Lean, J.（2004年）。太阳辐照度重建。载于数据贡献系列 *# 2004-035，IGBP PAGES/*世界古气候数据中心*/*美国国家海洋和大气管理局*/NGDC* 古气候项目，博尔德，科罗拉多州，美国。LeCun, Y.and Bengio, Y. and Hinton, G.（2015年）。深度学习。《自然》，521:436–444。Leibfried, F., Dutordoir, V., John, S., and Durrande, N.（2020年）。稀疏高斯过程与变分推断教程。*arXiv* 预印本 *arXiv:2012.13962*。Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., and Qiao, Y.（2024年）。VideoMamba：用于高效视频理解的状态空间模型。载于 *arXiv* 预印本 *arXiv:2403.06977*。Li, Y.（2018年）。近似推断：新视野。剑桥大学博士学位论文。Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M.（2023年）。用于生成建模的流匹配。载于第十一届国际学习表征会议。

Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T. 和 Lakshminarayanan, B.（2020）。通过距离感知，利用确定性深度学习进行简单且有理论依据的不确定性估计。载于神经信息处理系统进展。Liu, X., Gong, C. 和 Liu, Q.（2022）。流动又直又快：学习利用整流流生成和迁移数据。*arXiv* 预印本 *arXiv:2209.03003*。Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J. 和 Tang, J.（2021）。自监督学习：生成式还是对比式。*IEEE*知识与数据工程汇刊，35(1):857–876。Liu, Z., Luo, P., Wang, X. 和 Tang, X.（2015）。野外环境中的人脸属性深度学习。载于国际计算机视觉会议（*ICCV*）。Locatello, F., Bauer, S., Lucic, M., R¨atsch, G., Gelly, S., Sch¨olkopf, B. 和 Bachem, O.（2019）。对无监督学习解耦表征中的常见假设提出挑战。载于国际机器学习会议。Lou, A., Meng, C. 和 Ermon, S.（2024）。通过估计数据分布的比率进行离散扩散建模。载于国际机器学习会议（*ICML*）。L´azaro-Gredilla, M. 和 Figueiras-Vidal, A.（2009）。利用诱导特征进行稀疏推断的跨域高斯过程。载于神经信息处理系统进展。Ma, C. 和 Hern´andez-Lobato, J. M.（2021）。基于随机过程生成器的函数变分推断。载于神经信息处理系统进展。Ma, Y., Chen, T. 和 Fox, E.（2015）。随机梯度马尔可夫链蒙特卡罗的完整方案。载于神经信息处理系统进展。Maaløe, L., Fraccaro, M., Li´evin, V. 和 Winther, O.（2019）。Biva：用于生成建模的极深潜变量层级结构。神经信息处理系统进展，32。Maas, A. L., Daly, R. E., Pham, P. T., Huang, D.; Ng, A. Y. 和 Potts, C.（2011）。为情感分析学习词向量。载于 *NAACL-HLT*。Maddox, W. J., Stanton, S. 和 Wilson, A. G.（2021）。用于在线决策的稀疏变分高斯过程条件化。载于神经信息处理系统进展。

Matthews, B. W.（1975）。T4噬菌体溶菌酶二级结构预测结果与观测结果的比较。*Biochimica et Biophysica Acta (BBA)-ProteinStructure,405(2): 442–451*。McCulloch, W. S. 和 Pitts, W.（1943）。神经活动中内在观念的逻辑演算。数学生物物理学通报，5:115–133。Mercer, J.（1909）。正型与负型函数及其与积分方程理论的联系。英国皇家学会哲学汇刊，A(209):415–446。Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N. 和 Teller, A. H.（1953）。利用快速计算机进行状态方程计算。《化学物理学杂志》*21.6*。1087–1092。Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P. H. 和 Dokania, P. K.（2020）。利用焦点损失校准深度神经网络。载于神经信息处理系统进展。Naeini, M. P., Cooper, G. F. 和 Hauskrecht, M.（2015）。利用贝叶斯分箱获得校准良好的概率。*AAAI*人工智能会议论文集，29。Nair, V. 和 Hinton, G. E.（2010）。修正线性单元提升受限玻尔兹曼机的性能。载于国际机器学习会议。Nguyen, C. V., Li, Y., Bui, T. D. 和 Turner, R. E.（2018）。变分持续学习。载于国际学习表征会议。Nichol, A. 和 Dhariwal, P.（2021）。改进的去噪扩散概率模型。*arXiv* 预印本 *arXiv:2102.09672*。OpenAI, :, Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A. 等（2024a）。GPT-4o 系统卡。载于*arXiv* 预印本 *arXiv:2410.21276*。OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D. 等（2024b）。GPT-4 技术报告。载于 *arXiv* 预印本 *arXiv:2303.08774*。Papamarkou, T., Skoularidou, M., Palla, K., Aitchison, L., Arbel, J., Dunson, D., Filippone, M., Fortuin, V. 等（2024）。观点：在大规模人工智能时代，贝叶斯深度学习不可或缺。*arXiv* 预印本 *arXiv:2402.00809*。

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S.（2019年）。PyTorch：一种命令式风格的高性能深度学习库。载于神经信息处理系统进展第*32*卷，第8024–8035页。Curran Associates, Inc.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L.（2018年）。深度上下文化词表示。载于计算语言学协会北美分会年会。

Quan, C. 和 Li, X.（2024年）。面向静止和移动说话人的多通道长期流式神经语音增强。载于*arXiv* 预印本 *arXiv:2403.07675*。

Rahimi, A. 和 Recht, B.（2007年）。面向大规模核机器的随机特征。载于神经信息处理系统进展。

Rasmussen, C. E. 和 Williams, C. K. I.（2006年）。面向机器学习的高斯过程。麻省理工学院出版社。国际标准书号：0-262-18253-X。

Reid, I.、Choromanski, K.、Likhosherstov, V. 和 Weller, A.（2023年）。单纯形随机特征。载于国际机器学习会议。

Rezende, D. J.、Mohamed, S. 和 Wierstra, D.（2014年）。深度生成模型中的随机反向传播与近似推断。载于第*31*届国际机器学习会议论文集，第1278–1286页。

Rissanen, S.、Heinonen, M. 和 Solin, A.（2023年）。使用逆热耗散进行生成建模。国际学习表征会议。

Ritter, H.、Botev, A. 和 Barber, D.（2018年）。用于克服灾难性遗忘的在线结构化拉普拉斯近似。在神经信息处理系统进展，第31卷。

Ritter, H.、Kukla, M.、Zhang, C. 和 Li, Y.（2021年）。使用诱导权重在深度学习中进行稀疏不确定性表示。神经信息处理系统进展，34：6515–6528。

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N. 和 Aigrain, S.（2013年）。用于时间序列建模的高斯过程。《英国皇家学会哲学汇刊 *A*：数学、物理与工程科学》，*371(1984):20110550*。

Rombach, R., Blattmann, A., Lorenz, D., Esser, P. 和 Ommer, B.（2022年）。使用潜在扩散模型进行高分辨率图像合成。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第10684–10695页。

Rosenblatt, F. (1958). 感知机：大脑中信息存储与组织的概率模型。《心理学评论》，65:386。Rudin, W. (1994). 群上的傅里叶分析。*Wiley Classics Library. Wiley- Interscience* 纽约，重印版。Salimans, T. (2016年). 一种用于学习稀疏特征深层层次结构的结构化变分自编码器。*arXiv preprint arXiv:1602.08734*。

Salimans, T.、Kingma, D.与Welling, M.（2015年）。马尔可夫链蒙特卡罗与变分推断：弥合差距。载于国际机器学习会议（*ICML*）论文集。

Salimbeni, H.、Cheng, C.-A.、Boots, B.与Deisenroth, M.（2018年）。正交解耦变分高斯过程。载于 神经信息处理系统进展。Salimbeni, H.与Deisenroth, M.（2017年）。深度高斯过程的双重随机变分推断。载于神经信息处理系统进展。

S¨arkk¨a, S. 和 Solin, A.（2019年）。应用随机微分方程，第10卷。剑桥大学出版社。

Shi, Y., De Bortoli, V., Campbell, A., and Doucet, A. (2023). 扩散 Schrödinger 桥匹配。神经信息处理系统进展, 36. Shwartz Ziv, R. and LeC un, Y. (2024). 压缩还是不压缩——自监督学习与信息论：综述。《熵》，26(3):252. Snelson, E. and Ghahramani, Z. (2005). 使用伪输入的稀疏高斯过程。载于神经信息处理系统进展。Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). 使用非平衡热力学进行深度无监督学习。载于《机器学习国际会议》，页码 2256–2265。PMLR. Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. (2016). 阶梯式变分自编码器。神经信息处理系统进展, 29.

Song, J., Meng, C., and Ermon, S. (2021a). 去噪扩散隐式模型。载于国际学习表征会议。Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). 通过随机微分方程进行基于分数的生成建模。载于国际学习表征会议。Stanton, S., Maddox, W. J., Delbridge, I., and Wilson, A. G. (2021). 用于可扩展在线高斯过程的核插值。载于人工智能与统计国际会议。PMLR. Sun, S., Shi, J., Wilson, A. G., and Grosse, R. (2021). 通过谐波核分解实现可扩展的变分高斯过程。载于国际机器学习会议。Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). 函数变分贝叶斯神经网络。载于国际学习表征会议。Tfekci, P. and Kaya, H. (2014). 联合循环发电厂。UCI机器学习资料库。数字对象唯一标识符：https://doi.org/10.24432/C5002N。Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. (2020). 什么样的视图有利于对比学习? 神经信息处理系统进展, 33:6827–6839. Titsias, M. (2009). 稀疏高斯过程中诱导变量的变分学习。载于《人工智能与统计》。Ton, J.-F., Flaxman, S., Sejdinovic, D., and Bhatt, S. (2018). 利用高斯过程和非平稳傅里叶特征进行空间映射。空间统计学杂志, 28:59–78. Touvron, H., Cord, M., and Jegou, H. (2022). DeiT III：ViT的复仇。*arXiv*预印本 *arXiv:2204.07118*。Tran, D., Dusenberry, M. W., van der Wilk, M., and Hafner, D. (2019). 贝叶斯层：一种用于神经网络不确定性的模块。*arXiv:1812.03973*。Tran, G.-L., Milios, D., Michiardi, P., and Filippone, M. (2021). 利用邻居信息构建稀疏中的稀疏高斯过程。载于国际机器学习会议。Tsai, Y.-H. H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. (2019). Transformer剖析：通过核的视角统一理解Transformer的注意力。载于自然语言处理经验方法会议。

Vahdat, A. 和 Kautz, J.（2020）。NVAE：一种深度层次变分自编码器。神经信息处理系统进展，33:19667–19679。van den Oord, A.、Kalchbrenner, N.、Espeholt, L.、kavukcuoglu, K.、Vinyals, O. 和 Graves, A.（2016a）。使用PixelCNN解码器进行条件图像生成。载于神经信息处理系统进展。van den Oord, A.、Kalchbrenner, N. 和 kavukcuoglu, K.（2016b）。像素循环神经网络。载于国际机器学习会议。Van Den Oord, A.、Vinyals, O. 等（2017）。神经离散表示学习。神经信息处理系统进展，30。Van der Wilk, M.、Dutordoir, V.、John, S.、Artemev, A.、Adam, V. 和 Hensman, J.（2020）。域间和多输出高斯过程框架。*arXiv*预印本 *arXiv:2003.01115*。Vaswani, A.、Shazeer, N.、Parmar, N.、Uszkoreit, J.、Jones, L.、Gomez, A. N.、Kaiser, L. 和 Polosukhin, I.（2017）。注意力就是你所需的一切。神经信息处理系统进展，30。Villegas, R.、Babaeizadeh, M.、Kindermans, P.-J.、Moraldo, H.、Zhang, H.、Saffar, M. T.、Castro, S.、Kunze, J. 和 Erhan, D.（2022）。Phenaki：根据开放域文本描述生成变长视频。载于国际学习表征会议。Wang, F.-Y.、Yang, L.、Huang, Z.、Wang, M. 和 Li, H.（2024）。整流扩散：在整流流中，直线性并非必要。*arXiv*预印本 *arXiv:2410.07303*。Warstadt, A.、Singh, A. 和 Bowman, S. R.（2019）。神经网络可接受性判断。载于计算语言学协会汇刊。Wen, B.、Xu, C.、Wolfe, R.、Wang, L. L. 和 Howe, B.（2024）。缓解大语言模型中的过度自信：从行为视角看置信度估计与校准。载于*NeurIPS 2024*年行为机器学习研讨会。Wilkinson, W.、Solin, A. 和 Adam, V.（2021）。马尔可夫高斯过程的稀疏算法。载于人工智能与统计国际会议，第1747–1755页。PMLR。Wilson, A. G.、Hu, Z.、Salakhutdinov, R. 和 Xing, E. P.（2016）。深度核学习。载于人工智能与统计国际会议。Wilson, A. G. 和 Izmailov, P.（2022）。贝叶斯深度学习与泛化的概率视角。*arXiv*预印本 *arXiv:2002.08791*。

Wilson, J. T.、Borovitskiy, V.、Terenin, A.、Mostowsky, P. 和 Deisenroth, M. P.（2020）。高效地从高斯过程后验中采样函数。载于国际机器学习会议。Wu, M.、Zhuang, C.、Mosse, M.、Yamins, D. 和 Goodman, N.（2020）。关于视觉表征对比学习中的互信息。*arXiv*预印本 *arXiv:2005.13149*。Xiao, T. Z. 和 Bamler, R.（2023）。在层次变分自编码器的潜变量之间交换信息。载于第十一届国际学习表征会议。Xiong, M.、Hu, Z.、Lu, X.、Li, Y.、Fu, J.、He, J. 和 Hooi, B.（2024）。大语言模型能否表达其不确定性？对大语言模型置信度 elicitation 的实证评估。载于国际学习表征会议（*ICLR*）。Xue, B.、Yu, J.、Xu, J.、Liu, S.、Hu, S.、Ye, Z.、Geng, M.、Liu, X. 和 Meng, H.（2021）。用于语音识别的贝叶斯Transformer语言模型。*arXiv:2102.04754*。Yan, W.、Zhang, Y.、Abbeel, P. 和 Srinivas, A.（2021）。VideoGPT：使用VQ-VAE和Transformer生成视频。*arXiv*预印本 *arXiv:2104.10157*。Zeni, C.、Pinsler, R.、Z¨ugner, D.、Fowler, A.、Horton, M.、Fu, X.、Shysheya, S.、Crabb´e, J.、Sun, L.、Smith, J.、Nguyen, B.、Schulz, H.、Lewis, S.、Huang, C.-W.、Lu, Z.、Zhou, Y.、Yang, H.、Hao, H.、Li, J.、Tomioka, R. 和 Xie, T.（2024）。Mattergen：用于无机材料设计的生成模型。载于*arXiv*预印本 *arXiv:2312.03687*。Zhang, C.、B¨utepage, J.、Kjellstr¨om, H. 和 Mandt, S.（2018a）。变分推断进展。*IEEE*模式分析与机器智能汇刊，41(8):2008–2026。Zhang, R.、Isola, P.、Efros, A. A.、Shechtman, E. 和 Wang, O.（2018b）。深层特征作为感知度量的惊人效用。载于*IEEE*计算机视觉与模式识别会议论文集，第586–595页。Zhang, R.、Li, C.、Zhang, J.、Chen, C. 和 Wilson, A. G.（2019）。用于贝叶斯深度学习的循环随机梯度马尔可夫链蒙特卡罗。载于国际学习表征会议。Zhu, H.、Rodas, C. B. 和 Li, Y.（2023）。马尔可夫高斯过程变分自编码器。载于国际机器学习会议。

Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W. 和 Wang, X.（2024年）。视觉 Mamba：基于双向状态空间模型的高效视觉表征学习。载于国际机器学习会议 *(ICML)*。

附录 A

第 2章的补充材料

A.1 除 HiPPO-LegS 之外的 HiPPO 变体

除了 HiPPO-LegS，我们还在此提供了其他几种基于非均匀测度的 HiPPO 变体的公式（Gu 等人，2020年）。

HiPPO-LegT。 HiPPO-LegT 使用覆盖最近长度为 θ 的上下文窗口的均匀测度，$\omega^{(t)}(x) = \frac{1}{\theta}\mathbf{1}_{[t-\theta,t]}(x)$，并采用输入域适配于 $[t-\theta,t]$ 的缩放勒让德多项式，$g_m^{(t)}(x) = (2m+1)^{1/2}P_m\left(\frac{2(x-t)}{\theta}+1\right)$，作为基函数，其中 $P_m(\cdot)$ 是输入域为 $[-1,1]$ 的标准 m 阶勒让德多项式。相应线性常微分方程的矩阵 $\boldsymbol{A}(t) \in \mathbb{R}^{M\times M}$ 和 $\boldsymbol{B}(t) \in \mathbb{R}^{M\times 1}$ 的公式为：

$$\boldsymbol{A}(t) = -\frac{1}{\theta}\boldsymbol{A}, \quad [\boldsymbol{A}]_{mk} = \begin{cases} \sqrt{(2m+1)(2k+1)} & \text{if } m \geq k \\ (-1)^{m-k}\sqrt{(2m+1)(2k+1)} & \text{if } m < k \end{cases} \tag{A.1}$$

and

$$[\boldsymbol{B}(t)]_m = \frac{[\boldsymbol{B}]_m}{\theta} = \frac{\sqrt{2m+1}}{\theta}, \tag{A.2}$$

HiPPO-LagT。 HiPPO-LagT 使用指数衰减测度，对近期历史序列赋予更高的重要性，$\omega^{(t)}(x) = \exp(-(t-x))\ \mathbf{1}_{(-\infty,t]}(x)$，并使用输入域适配于 $(-\infty,t]$，$g_m^{(t)}(x) = L_m(t-x)$ 的缩放拉盖尔多项式作为基函数，其中 $L_m(\cdot)$ 是输入域为 $[0,\infty$ 的标准 m 阶拉盖尔多项式。矩阵 $\boldsymbol{A}(t) \in \mathbb{R}^{M\times M}$ 的公式以及

$\boldsymbol{B}(t) \in \mathbb{R}^{M\times 1}$ 对应线性常微分方程的公式如下：

$$[\boldsymbol{A}(t)]_{mk} = [\boldsymbol{A}]_{mk} = \begin{cases} -1 & \text{if } m \geq k \\ 0 & \text{if } m < k \end{cases} \tag{A.3}$$

and

$$[\boldsymbol{B}(t)]_m = [\boldsymbol{B}]_m = 1, \tag{A.4}$$

HiPPO-FouT。与 HiPPO-LegT 类似，HiPPO-FouT 也在滑动窗口 $\omega^{(t)}(x) = \frac{1}{\theta}\mathbf{1}_{[t-\theta,t]}(x)$ 上采用均匀测度，但它基于缩放傅里叶基函数，并将输入域调整为 $[t-\theta, t]$, $g_m^{(t)}(x) = \exp\left(2\pi i m \frac{t-x}{\theta}\right)$，作为基函数。相应线性常微分方程的矩阵 $\boldsymbol{A}(t) \in \mathbb{R}^{M\times M}$和$\boldsymbol{B}(t) \in \mathbb{R}^{M\times 1}$ 的公式如下：

$$[\boldsymbol{A}(t)]_{mk} = [\boldsymbol{A}]_{mk} = \begin{cases} -\frac{1}{\theta} & \text{if } m \neq k \\ \frac{2\pi i m - 1}{\theta} & \text{if } m = k \end{cases} \tag{A.5}$$

and

$$[\boldsymbol{B}(t)]_m = [\boldsymbol{B}]_m = \frac{1}{\theta}, \tag{A.6}$$

附录 B

第 3 章的补充材料

B.1 推导

B.1.1 稀疏变分高斯过程的证据下界推导

为完整起见，我们给出标准稀疏变分高斯过程（Titsias，2009年；Hensman 等人，2013年）的证据下界完整推导，如式2.38 所示。设有 M 诱导点对$\{(\boldsymbol{z}_m,\ u_m)\}_{m=1}^{M}$，则 $[\boldsymbol{f},\boldsymbol{u}]^\top$ 的先验分布为：

$$p(\boldsymbol{f},\boldsymbol{u}|\boldsymbol{X},\boldsymbol{Z})=\mathcal{N}(\mathbf{0},\begin{pmatrix}\boldsymbol{K}_{\boldsymbol{XX}} & \boldsymbol{K}_{\boldsymbol{XZ}}\\ \boldsymbol{K}_{\boldsymbol{ZX}} & \boldsymbol{K}_{\boldsymbol{ZZ}}\end{pmatrix}). \tag{B.1}$$

在先验条件匹配假设下（见式 2.35），给定诱导点 $\boldsymbol{u}$，对于输入 $\boldsymbol{X}$，函数值的近似后验条件分布 $\boldsymbol{f}$ 与先验条件分布相同：

$$q(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X})=p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X}). \tag{B.2}$$

在先验条件匹配假设下，$q(\boldsymbol{f},\ \boldsymbol{u}|\boldsymbol{Z},\ \boldsymbol{X})=p(\boldsymbol{f}|\boldsymbol{u},\ \boldsymbol{Z},\ \boldsymbol{X})q(\boldsymbol{u})$，其中 $q(\boldsymbol{u})=\mathcal{N}(\boldsymbol{m}_{\boldsymbol{u}},\ \boldsymbol{S}_{\boldsymbol{u}})$。假设观测似然为 $p(\boldsymbol{y}|\boldsymbol{f})$，证据下界可以

可简化如下：

$$\begin{aligned}\mathcal{L}_{ELBO} &= \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u}|\boldsymbol{Z},\boldsymbol{X})}[\log \frac{p(\boldsymbol{y},\boldsymbol{f},\boldsymbol{u}|\boldsymbol{Z},\boldsymbol{X})}{q(\boldsymbol{f},\boldsymbol{u}|\boldsymbol{Z},\boldsymbol{X})}] \\ &= \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u}|\boldsymbol{Z},\boldsymbol{X})}[\log \frac{p(\boldsymbol{y}|\boldsymbol{f})\cancel{p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X})}p(\boldsymbol{u}|\boldsymbol{Z})}{\cancel{p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X})}q(\boldsymbol{u})}] \\ &= \int (\int p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X})q(\boldsymbol{u})d\boldsymbol{u}) \log p(\boldsymbol{y}|\boldsymbol{f})d\boldsymbol{f} \\ &\qquad + \int q(\boldsymbol{u}) \log \frac{p(\boldsymbol{u}|\boldsymbol{Z})}{q(\boldsymbol{u})} \cancel{\int p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{Z},\boldsymbol{X})d\boldsymbol{f}}d\boldsymbol{u} \\ &= \underbrace{\mathbb{E}_{q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{Z})}[\log p(\boldsymbol{y}|\boldsymbol{f})]}_{\text{ELL}} - \underbrace{\mathrm{KL}(q(\boldsymbol{u})||p(\boldsymbol{u}|\boldsymbol{Z}))}_{\text{KL regularizer}}.\end{aligned} \tag{B.3}$$

这里，$q(\boldsymbol{f}|\boldsymbol{X},\ \boldsymbol{Z}) = \int p(\boldsymbol{f}|\boldsymbol{u},\ \boldsymbol{Z},\ \boldsymbol{X})q(\boldsymbol{u})d\boldsymbol{u}$ 是一个高斯分布，表示为：

$$q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{Z}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{XZ}}\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{m}_{\boldsymbol{u}}, \boldsymbol{K}_{\boldsymbol{XX}} + \boldsymbol{K}_{\boldsymbol{XZ}}\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}(\boldsymbol{S}_{\boldsymbol{u}} - \boldsymbol{K}_{\boldsymbol{ZZ}})\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{K}_{\boldsymbol{ZX}}). \tag{B.4}$$

采用高斯似然时，证据下界中的第一项可以解析求得。否则，我们使用蒙特卡洛样本 $\boldsymbol{f} \sim q(\boldsymbol{f}|\boldsymbol{X},\ \boldsymbol{Z})$ 来估计第一项。第二项是两个高斯分布之间的 KL 散度，因此可以写成封闭形式：

$$\mathrm{KL}(q(\boldsymbol{u})||p(\boldsymbol{u}|\boldsymbol{Z})) = \frac{1}{2}\left[Tr(\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{S}_{\boldsymbol{u}}) + \boldsymbol{m}_{\boldsymbol{u}}^{\top}\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{m}_{\boldsymbol{u}} + \text{对数}\ \frac{|\boldsymbol{K}_{\boldsymbol{ZZ}}|}{|\boldsymbol{S}_{\boldsymbol{u}}|}\right] + \text{常数。} \tag{B.5}$$

在标准 SGPA 中，证据下界目标函数基本保持不变，只是由于变分均值重新参数化为 $\boldsymbol{v} := \boldsymbol{K}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{m}_{\boldsymbol{u}}$，$q(\boldsymbol{f}|\boldsymbol{X},\ \boldsymbol{Z})$ 的均值变为 $\boldsymbol{K}_{\boldsymbol{XZ}}\boldsymbol{v}$，而 $\mathrm{KL}(q(\boldsymbol{u})||p(\boldsymbol{u}|\boldsymbol{Z}))$ 中的二次项变为 $\boldsymbol{v}^{\top}\boldsymbol{K}_{\boldsymbol{ZZ}}\boldsymbol{v}$。

B.1.2 基于 SGPA 的 Transformer 证据下界推导

一个 L层的、基于 SGPA 的 Transformer 是一个深度 GP（Damianou 和 Lawrence，2013年），我们使用双重随机变分推断框架（Salimbeni 和 Deisenroth，2017年）对其进行训练。对于每个输入序列 $\boldsymbol{F}^0 := \boldsymbol{X}$，$\boldsymbol{Y}$、$\{\boldsymbol{F}^l\}_{l=1}^{L}$、$\{\boldsymbol{u}^{l,\ h}_{a\cup g}\}_{l=1,\ h=1}^{L\ H}$ 的联合分布为：

$$\begin{aligned}&p(\boldsymbol{Y},\{\boldsymbol{F}^l\}_{l=1}^{L},\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{l=1,h=1}^{L,H}|\boldsymbol{F}^0) = \\ &\qquad p(\boldsymbol{Y}|\boldsymbol{F}^L)[\prod_{l=1}^{L} p(\boldsymbol{F}^l|\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{h=1}^{H},\boldsymbol{F}^{l-1})p(\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{h=1}^{H}|\{\boldsymbol{k}_{g}^{l,h}\}_{h=1}^{H},\boldsymbol{F}^{l-1})],\end{aligned} \tag{B.6}$$

其中 $p(\{\boldsymbol{u}^{l,\ h}_{,a\cup g}\}_{h=1}^{H}|\{\boldsymbol{k}^{l,\ h}_{,g}\}_{h=1}^{H},\boldsymbol{F}^{l-1}) = \prod_{h=1}^{H} p(\boldsymbol{u}^{l,\ h}_{,\ a\cup g}|\boldsymbol{k}^{l,\ h}_{,g},\boldsymbol{F}^{l-1})$，这是因为我们假设每一层中诱导点的先验在各个注意力头之间可分解。请注意，此处摊销键 $\boldsymbol{k}^{l,\ h}_{,a}$ 以确定性方式依赖于 $\boldsymbol{F}^{l-1}$，因此我们省略了摊销

条件化中的键项。假设先验条件匹配，即，

$$q(\boldsymbol{F}^l|\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{h=1}^H,\boldsymbol{F}^{l-1})=p(\boldsymbol{F}^l|\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{h=1}^H,\boldsymbol{F}^{l-1}), \tag{B.7}$$

对于$\{\boldsymbol{F}^l\}_{l=1}^L$，$\{\boldsymbol{u}^{l,\ h}_{a\cup g}\}_{l=1,\ h=1}^{L\ H}$的联合近似后验为：

$$q(\{\boldsymbol{F}^l\}_{l=1}^L,\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{l=1,h=1}^{L,H}|\boldsymbol{F}^0)=\prod_{l=1}^L p(\boldsymbol{F}^l|\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{h=1}^H,\boldsymbol{F}^{l-1})q(\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{h=1}^H|\{\boldsymbol{k}_g^{l,h}\}_{h=1}^H,\boldsymbol{F}^{l-1}), \tag{B.8}$$

其中，由于我们令$\{\boldsymbol{u}^{l,\ h}_{a\cup g}\}_{h=1}^H$的近似分布也在注意力头之间进行因子分解，因此$q(\{\boldsymbol{u}^{l,\ h}_{a\cup g}\}_{h=1}^H|\{\boldsymbol{k}^{l,\ h}_{g}\}_{h=1}^H,\boldsymbol{F}^{l-1})=\prod_{h=1}^H q(\boldsymbol{u}^{l,\ h}_{a\cup g}|\boldsymbol{k}^{l,\ h}_{g},\boldsymbol{F}^{l-1})$。证据下界的推导方式与单层高斯过程的情况（公式 B.3）类似，并且 q 和 p 中的条件分布项再次相互抵消。这将证据下界简化为：

$$\begin{aligned}\mathcal{L}_{ELBO}&=\mathbb{E}_{q(\{\boldsymbol{F}^l\}_{l=1}^L,\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{l=1,h=1}^{L,H}|\boldsymbol{F}^0)}[\log\frac{p(\boldsymbol{Y},\{\boldsymbol{F}^l\}_{l=1}^L,\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{l=1,h=1}^{L,H}|\boldsymbol{F}^0)}{q(\{\boldsymbol{F}^l\}_{l=1}^L,\{\boldsymbol{u}_{a\cup g}^{l,h}\}_{l=1,h=1}^{L,H}|\boldsymbol{F}^0)}]\\&=\mathbb{E}_{q(\boldsymbol{F}^L|\boldsymbol{F}^0,\{\boldsymbol{k}_g^{l,h}\}_{l=1,h=1}^{L,H})}[\log p(\boldsymbol{Y}|\boldsymbol{F}^L)]\\&\quad-\sum_{l=1}^L\sum_{h=1}^H\mathbb{E}_{q(\boldsymbol{F}^l|\boldsymbol{F}^0,\{\boldsymbol{k}_g^{j,h}\}_{j=1,h=1}^{l,H})}\{\mathrm{KL}(q(\boldsymbol{u}_{a\cup g}^{l,h}|\boldsymbol{k}_g^{l,h},\boldsymbol{F}^{l-1})||p(\boldsymbol{u}_{a\cup g}^{l,h}|\boldsymbol{k}_g^{l,h},\boldsymbol{F}^{l-1}))\},\end{aligned} \tag{B.9}$$

其中

$$\begin{aligned}&q(\boldsymbol{F}^l|\boldsymbol{F}^0,\{\boldsymbol{k}_g^{j,h}\}_{j=1,h=1}^{l,H})=\\&\qquad\int\prod_{j=1}^l p(\boldsymbol{F}^j|\{\boldsymbol{u}_{a\cup g}^{j,h}\}_{h=1}^H,\boldsymbol{F}^{j-1})q(\{\boldsymbol{u}_{a\cup g}^{j,h}\}_{h=1}^H|\{\boldsymbol{k}_g^{j,h}\}_{h=1}^H,\boldsymbol{F}^{j-1})d\boldsymbol{u}_{a\cup g}^{1:l,1:H}d\boldsymbol{F}^{1:l-1}.\end{aligned} \tag{B.10}$$

证据下界中的两项均可利用通过重参数化技巧在每一层迭代生成的样本进行估计。对于第二项“正则化”项，期望内的 KL 散度具有简化形式，因为我们假设对每个注意力输出维度（d）拟合一个独立的解耦 SVGP：

$$\begin{aligned}&\mathrm{KL}(q(\boldsymbol{u}_{a\cup g}^{l,h}|\boldsymbol{k}_g^{l,h},\boldsymbol{F}^{l-1})||p(\boldsymbol{u}_{a\cup g}^{l,h}|\boldsymbol{k}_g^{l,h},\boldsymbol{F}^{l-1}))\\&=\frac{1}{2}\sum_{d=1}^D\{[\boldsymbol{v}_a^{l,h}]_{:,d}^\top(\boldsymbol{K}_{\boldsymbol{k}_a^{l,h}\boldsymbol{k}_a^{l,h}}-\boldsymbol{K}_{\boldsymbol{k}_a^{l,h}\boldsymbol{k}_g^{l,h}}\boldsymbol{K}_{\boldsymbol{k}_g^{l,h}\boldsymbol{k}_g^{l,h}}^{-1}\boldsymbol{K}_{\boldsymbol{k}_g^{l,h}\boldsymbol{k}_a^{l,h}})[\boldsymbol{v}_a^{l,h}]_{:,d}+[\boldsymbol{v}_g^{l,h}]_{:,d}^\top\boldsymbol{K}_{\boldsymbol{k}_g^{l,h}\boldsymbol{k}_g^{l,h}}[\boldsymbol{v}_g^{l,h}]_{:,d}\\&\quad+\mathrm{Tr}([\boldsymbol{S}_g^{l,h}]_{:,:,d}\boldsymbol{K}_{\boldsymbol{k}_g^{l,h}\boldsymbol{k}_g^{l,h}}^{-1})-\log\det[\boldsymbol{S}_g^{l,h}]_{:,:,d}+\log\det\boldsymbol{K}_{\boldsymbol{k}_g^{l,h}\boldsymbol{k}_g^{l,h}}\}+\mathrm{const},\end{aligned} \tag{B.11}$$

其中 D 是注意力输出维度的总数， M_g 是每个头的全局诱导点数量。

B.2 不确定性校准指标

我们使用两种广泛应用的指标来评估分布内数据的不确定性估计质量：适当评分规则和校准误差。具体而言，负对数似然是一种基于适当评分规则的指标，而期望校准误差（ECE）和最大校准误差（MCE）是两种基于校准误差的指标。

B.2.1 适当评分规则

适当评分规则（Gneiting 等，2007）是一个以分布和数据点为输入的函数。假设数据生成分布为 p_d，模型给出的预测分布为 $\hat{p}$，那么对于每个观测 $\boldsymbol{x}_i$ ，我们都可以基于预测分布得到一个得分 $S(\hat{p}, \boldsymbol{x}_i)$。为了衡量 $\hat{p}$ 与 p_d 之间的偏差，我们对 p_d 上述评分取期望： $S(\hat{p}, p_d) = \mathbb{E}_{p_d}[S(\hat{p}, \boldsymbol{x})]$。对于适当评分规则， $S(p_d, p_d) \leq S(\hat{p}, p_d)$， $\forall \hat{p}$，因此当 $\hat{p} = p_d$ 时， $S(\hat{p}, p_d)$ 可达到最小值。对于严格适当的评分规则， $S(p_d, p_d)$ 是唯一的全局最小值。这说明了为何适当评分规则可用于评估预测分布的质量。在实践中，由于真实值 p_d 未知， $S(\hat{p}, p_d)$ 在解析上难以求解，因此我们通常采用蒙特卡洛方法进行估计。负对数似然（NLL）是一种基于适当评分规则（具体而言是对数评分）计算得出的常用指标。给定测试集 $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{n=1}^{N}$，NLL 的计算方式如下：

$$\mathrm{NLL} = -\mathbb{E}_{p_d}[\log \hat{p}(\boldsymbol{y}|\boldsymbol{x})] \approx -\frac{1}{N}\sum_{n=1}^{N} \log \hat{p}(\boldsymbol{y}_n \mid \boldsymbol{x}_n).$$

B.2.2 校准误差

校准误差（Naeini 等人，2015年；Guo 等人，2017年）衡量模型对分类正确性的置信度与正确性真实概率之间的偏差。数据点被分类到的类别的预测概率 $\hat{p}$，反映了模型对分类正确性的置信度。对于校准良好的模型，在每个置信度水平 $\hat{p}$下，比例 p （即属于该“$\hat{p}$-置信度组”且被正确分类的数据点所占比例）应等于 $\hat{p}$。随后基于 $\hat{p}$ 与 p之间的绝对差计算校准误差。期望校准误差是该差值关于

置信度$\hat{}p$分布的期望：

$$ECE = \mathbb{E}_{\hat{p}}\left[|\hat{p} - p|\right] \tag{B.12}$$

MCE 是这些差值中的最大值：

$$MCE = \arg\max_{\hat{p}} |\hat{p} - p| \tag{B.13}$$

在实践中，由于每个置信度水平下的数据量不足，使用直方图分箱会得到有偏的校准误差估计（Naeini 等人，2015年；Guo 等人，2017年）。我们将 $\hat{}p$、$[0, 1]$ 的定义域划分为 15 个大小相等的区间，与 Guo 等人 (2017 年) 中的方法相同。

B.3 补充实验

B.3.1 标准 SGPA 与解耦 SGPA 的实证比较

在初步实验中，我们比较了基于标准 SGPA 与解耦 SGPA 的 ViT 在 CIFAR10 图像分类任务上的性能，且未使用数据增强。我们还考虑了基于 DSVGP（Cheng 和 Boots，2017年）的解耦 SGPA。根据公式 3.10，基于 DSVGP（Cheng 和 Boots，2017年）的解耦 SGPA 的后验均值和协方差公式如下：

$$\begin{aligned} \boldsymbol{m}_d &= \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}_a}[\boldsymbol{v}_a]_{:,d}, \\ \boldsymbol{\Sigma}_d &= \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{q}\boldsymbol{k}_g}\boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{k}_g}^{-1}([\boldsymbol{S}_g]_{:,:,d} - \boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{k}_g})\boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{k}_g}^{-1}\boldsymbol{K}_{\boldsymbol{k}_g\boldsymbol{q}}, \end{aligned} \tag{B.14}$$

表 B.1 显示，基于标准 SGPA 的 ViT 和基于 DSVGP 的解耦 SGPA（Cheng 和 Boots，2017年）的表现不如基于 ODSVGP 的解耦 SGPA（Salimbeni 等人，2018年）。具体而言，标准 SGPA 对数据存在明显欠拟合。因此，在其余实验中，我们仅考虑基于 ODSVGP 的解耦 SGPA（Salimbeni 等人，2018年）。

表 B.1：基于标准 SGPA 和解耦 SGPA 的两种变体的视觉变换器在未使用数据增强的 CIFAR10 上训练后的测试集准确率和负对数似然。

模型	准确率	NLL
标准 SGPA	0.6435±0.0039	1.0159±0.0065
解耦 SGPA（Cheng 和 Boots，2017年）	0.7513±0.0014	0.7877±0.0045
解耦 SGPA（Salimbeni 等人，2018年）	**0.7787±0.0024**	**0.6968±0.0032**

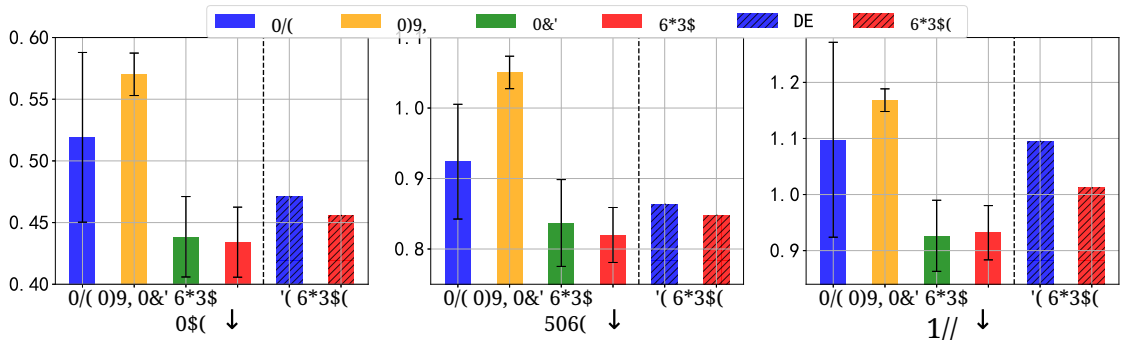


图 B.1：在 ZINC 上训练的 Transformer 的测试集回归误差和负对数似然指标。

B.3.2 使用 ZINC数据集进行图属性回归

对于图属性回归，我们假设观测似然服从拉普拉斯分布，其尺度参数可训练 b（即，观测似然的密度为 $g(y|f) = \frac{1}{2b}\exp(-\frac{|y-f|}{b})$，其中 f 是 Transformer 输出的标量函数值）。我们计算平均绝对误差（MAE）、均方根误差（RMSE）和负对数似然（NLL），以评估模型，结果如图 B.1所示。此外，我们将预测方差用作得分，并在图 B.2中评估分布外检测性能。请注意，在这种情况下，最大似然估计对分布外检测没有作用，因为它对所有实例产生相同的预测方差。我们使用由测试集生成的合成分布外数据集：对于每个测试实例，从邻接矩阵中移除现有边，并在原本没有连接的节点之间添加边。在“单模型”方法中，SGPA 和 MCD 的结果远优于最大似然估计和平均场变分推断。对于这项任务，SGPA 与 MCD 的性能差异可以忽略不计。不过，与 MCD 相比，SGPA 的表现更稳健，因为它给出的误差条更小。集成方法在分布外检测方面优于“单模型”方法，其中 SGPAE 的结果最佳。

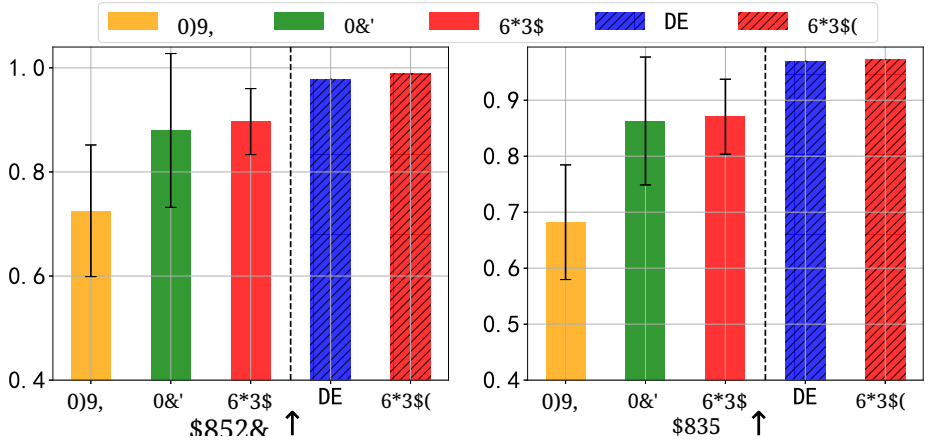


图 B.2：使用在 ZINC 上训练的 Transformer 进行分布外检测的 AUROC 和 AUPR 指标。

有趣的是，在分布内校准方面，他们的表现不如 SGPA 和 MCE。

B.4 详细实验设置与超参数

各项实验共用的训练设置。对于最大似然估计和 MCD，我们最初考虑了 0.1 和 0.2 两种 Dropout 率，但在图像分类的初步实验中发现，使用 0.1 Dropout 率训练的模型在准确率方面始终优于使用 0.2 Dropout 率训练的模型。因此，除 MFVI 外，我们决定所有方法均使用 0.1 的 Dropout 率。每一层均采用均值池化策略。每个注意力层中的非线性映射 G_{ϕ^l} 采用与Vaswani 等人（2017年）相同的方式，以一个 2 层多层感知机进行参数化。对于采用基于核的注意力的模型，情感分析和语言可接受性任务使用指数核（Tsai 等人，2019年）： $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp(\sum_{j=1}^{D} \frac{x_j x_j'}{\sigma_j^2})$；图像分类和图属性回归任务使用 ARD-RBF 核（Rasmussen 和 Williams，2006年）： $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp(-\frac{1}{2} \sum_{j=1}^{D} \frac{(x_j - x_j')^2}{\sigma_j^2})$，其中 D 是 $\boldsymbol{x}$ 的维度，$\boldsymbol{x}'$、 σ_f^2是输出方差， σ_j是第 j 个维度的长度尺度。对于 MFVI、MCD、SNGP 和 SGPA，使用 10 个蒙特卡洛样本估计预测不确定性。

- **初始化：** 我们从头开始训练所有模型（即所有参数均随机初始化，不进行任何预训练）。除全局诱导点参数外，其余参数均与标准 Transformer 中的相同，并采用深度学习平台的默认方法进行初始化（我们使用 PyTorch（Paszke 等人，2019年））。全局诱导位置和全局变分均值的每个维度均采用标准高斯分布进行随机初始化。对于用于参数化全局变分协方差的 Cholesky 因子，下三角部分中的每个元素以及对数对角线中的每个元素均采用标准高斯分布进行随机初始化。

- **优化：** 所有模型均使用 ADAM 优化器进行训练（Kingma 和 Ba，2015年），对于一个批次中的每个输入序列，我们只抽取一个样本来估计证据下界（公式 3.18）。在实验中，我们没有观察到优化问题：训练过程中，证据下界始终稳定下降，没有明显的尖峰。

使用 IMDB 进行情感分析（Maas 等人，2011年）。我们考虑 5 种不同的数据划分，每种划分包含 35,000 个训练样本、5,000 个验证样本和 10,000 个测试样本。每个输入序列的词元数上限为 512。在架构方面，我们使用

一个包含 1 个 MHSA 层和 8 个注意力头的 Transformer，嵌入维度和隐藏维度均为 128。对于 SGPA，我们为每个头使用 50 个全局诱导点。除事后方法、TS 和 KFLLLA 外，我们使用批次大小 32、初始学习率 0.001（线性衰减至 0.0001），对所有模型训练 20 个训练轮次。我们根据每个训练轮次结束后计算的验证准确率选择最佳模型。

使用 CoLA 评估语言可接受性（Warstadt等人，2019年）。使用原始数据集提供的 516 个分布外样本评估模型的分布外鲁棒性。在 5 次独立运行中，将剩余的 9,078 个分布内样本随机划分为 7,262 个训练样本和 1,816 个分布内测试样本。我们使用一个包含 2 个 MHSA 层的 Transformer，每层有 4 个注意力头，嵌入维度为 128，隐藏维度为 256。输入嵌入采用 ELMo 风格表示（Peters等人，2018年）。对于 SGPA，每个注意力头使用 5 个全局诱导点。除事后方法 KFLLLA 外，我们对所有模型训练 50 轮，批次大小为 32，初始学习率为 0.0005，并线性衰减至 0.00001；评估时使用最后一轮训练得到的模型。

使用 CIFAR10 和 CIFAR100 进行图像分类（Krizhevsky 等人，2009年）。对于 CIFAR10 和 CIFAR100 数据集，我们将原始训练集随机划分为 4,5000 个训练样本和 5,000 个验证样本，并在原始的 10,000 个测试样本上进行测试。输入图像按图像块大小为4×4进行分词。对于不使用数据增强的 CIFAR10，我们采用一个包含 5 个 MHSA 层的 ViT（Dosovitskiy 等人，2021年），每层包含 4 个注意力头，隐藏维度为 128。对于所有其他实验，我们使用一个包含 6 层、4 个注意力头且隐藏维度为 256 的 ViT。除事后方法、TS 和 KFLLLA 外，我们对所有模型（SGPA 除外）训练 600 个训练轮次，批次大小为 100，初始学习率为 0.0005，并线性衰减至 0.00001。对于 SGPA，我们为每个头使用 32 个全局诱导点，并使用最大似然估计第 100 个训练轮次的参数初始化深度核超参数，然后继续训练 500 个训练轮次。根据每 10 个训练轮次计算一次的验证准确率选出最佳模型。对于使用数据增强的实验，我们采用与Touvron 等人（2022年）相同的数据增强策略（3-Augment）。

使用 ZINC 进行图属性回归（Dwivedi 等人，2020年）。结果见B.3.2 ，这些结果取自 3 次独立运行的平均值。我们采用与（Dwivedi 等人，2020年）相同的数据划分，因此包含 10,000 个训练样本、1,000 个验证样本和 1,000 个测试样本。我们没有对网络架构进行图特定修改，而是采用Kim 等人（2022年）提出的特征工程技术，将每个图转换为输入嵌入序列。我们考虑

采用 8 层、8 个注意力头的 Transformer，嵌入维度和隐藏维度均为 80。对于 SGPA，每个注意力头使用 10 个全局诱导点。所有模型均训练 500 轮，批次大小为 64，初始学习率为 0.0004，并线性衰减至 0.000002。我们根据每 10 轮结束时计算的验证准确率选择最佳模型。

B.5 运行时间

我们在此分析 SGPA 的墙钟训练与推理时间。在表 B.2 中，我们给出推断阶段单个批次在 10 个蒙特卡洛样本下的计算时间，对象为 CoLA（批次大小 $= 227$）和 CIFAR10 （批次大小 $= 200$ ）（结果由单张 Nvidia GTX 2080 Ti 图形处理器卡获得）。对于 SGPA，我们首先付出一次性成本，对与全局诱导点相关的核矩阵求逆（$\boldsymbol{K}_{\boldsymbol{k}_g \boldsymbol{k}_g}^{-1}$）。完成之后，我们将它们视为常数矩阵并代入式 3.15。在生成传递到下一层的样本时，我们将协方差（$\boldsymbol{\Sigma}_d$）进行对角化（式 3.15 ），以避免高成本地计算 $\boldsymbol{\Sigma}_d$ 的 Cholesky 因子。

计算成本取决于所使用的全局诱导点数量。对于 CoLA，我们每个头仅使用了 5 个全局诱导点，MCD 与 SGPA 之间推理时间的相对差异小于 CIFAR10 的情况，后者每个头使用 32 个全局诱导点。值得注意的是，我们并未针对全局诱导点数量进行广泛的超参数调优。SGPA 很可能在更少的全局诱导点下仍然表现良好。例如，对于 CIFAR10，我们后来还训练了一个每个头使用 16 个全局诱导点的 SGPA 模型，并未观察到明显的性能下降（准确率：0.7790，负对数似然：0.7259，期望校准误差：0.0625，最大校准误差：0.0819）。在这种情况下，推理时间可从 0.986 进一步降至 0.807。

表 B.2：在推断阶段，使用 10 个蒙特卡洛样本处理 CoLA（批次大小为 $= 227$）和 CIFAR10（批次大小为 $= 200$）时，单个批次所需的计算时间（单位： s）（结果使用单张 Nvidia GTX 2080 Ti 图形处理器显卡得出）。

	MCD	平均场变分推断	SGPA
CoLA	1.839	1.882	2.358
CIFAR10（32 $\boldsymbol{k}_g$）	0.234	0.395	0.986
CIFAR10（16 $\boldsymbol{k}_g$）	0.234	0.395	0.807

在表 B.3 中，我们展示了 SGPA 和最大似然估计在 CoLA（批次大小 s）以及 CIFAR10（批次大小 $= 100$）上每个训练轮次的训练时间（单位为 $= 32$）（结果使用单张 Nvidia GTX 2080 Ti 图形处理器卡获得）：

表 B.3：SGPA 和最大似然估计在 CoLA（批次大小 $= 32$）和 CIFAR10（批次大小 $= 100$）上一个训练轮次的训练时间（单位： s）（结果使用单张 Nvidia GTX 2080 Ti 图形处理器卡获得）。

	MLE	SGPA
CoLA	17.834	20.089
CIFAR10（32 $\boldsymbol{k}_g$）	30.593	138.609
CIFAR10（16 $\boldsymbol{k}_g$）	30.593	109.298

B.6 与稀疏中的稀疏高斯过程的联系

与标准稀疏高斯过程不同，后者的诱导点在所有输入之间共享；SGPA 由一组依赖于输入（或称“摊销”）的诱导位置 $\{\boldsymbol{k}^{l,\ h}_{a}\}$及其对应的变分参数组成，这意味着我们为每个输入序列使用不同的均值函数。因此，基于 SGPA 的 Transformer 无法完美地对输入序列之间的相关性建模。它们只能为每个输入序列提供边际不确定性。尽管如此，我们通过实证发现，在文本或图像分类等应用中，相关性可能并不重要。

有趣的是，SGPA 可被视为稀疏中的稀疏高斯过程（SWSGP）（Tran 等人，2021年；Jafrasteh 等人，2022年）的一种实例化，它允许为每个输入使用自适应诱导点。假设指标集（在我们的情形中为嵌入空间）为 $\boldsymbol{\chi}$，且 $p(\boldsymbol{Z})$ 是 $\boldsymbol{\chi}$ 的 M 元子集上的分布（即每次随机抽取都会从 $\boldsymbol{\chi}$ 中给出 M 个诱导位置）。联合先验和近似后验分别变为 $p(\boldsymbol{f}, \boldsymbol{u}, \boldsymbol{Z}) = p(\boldsymbol{f}|\boldsymbol{u})p(\boldsymbol{u}|\boldsymbol{Z})p(\boldsymbol{Z})$ 和 $q(\boldsymbol{u}, \boldsymbol{Z}) = q(\boldsymbol{u}|\boldsymbol{Z})p(\boldsymbol{Z})$。在 Tran 等人（2021年）中，对于每个输入 $\boldsymbol{x}$，他们提议从训练输入中选取其 M 个最近邻作为诱导位置，从而使 $q(\boldsymbol{Z})$ 成为以 $\boldsymbol{x}$ 为条件的狄拉克分布，而 $q(\boldsymbol{u}|\boldsymbol{Z})$ 则是在所选诱导位置处所评估函数值上的边际变分分布。相比之下，对于每个输入序列 $\boldsymbol{x}$，在层 l 中，基于 SGPA 的 Transformer 所使用的诱导位置既包括依赖于输入的诱导位置（$\{\boldsymbol{k}^{l,}_{a}{}^{h}\}_{h=1}^{H}$），它们如 Jafrasteh 等人（2022年）那样通过神经网络从 $\boldsymbol{x}$ 获得，也包括全局诱导位置 $\{\boldsymbol{k}^{l,}_{g}{}^{h}\}_{h=1}^{H}$，这些位置在所有输入序列之间共享。

请注意，测试时使用的依赖于输入的诱导点可能在训练期间并未出现。相反，我们依赖所学习的神经网络对其进行摊销（Jafrasteh 等人，2022年）。因此，当测试序列包含与训练序列中的词元相距甚远的词元时，拟合的均值函数可能不一致（如 $\boldsymbol{v}_a$ 可能不一致）。根据经验，我们发现，对于分布内测试序列，这种不一致问题并不严重。此外，我们认为，对于分布外输入，尽管拟合的后验均值可能不可靠，但不确定性

仍会增加，因为后验协方差完全由全局诱导点决定，而全局诱导点不存在不一致问题。直观地说，在所有输入序列之间共享的全局键 $\{\boldsymbol{k}^{l,\ h}_{g}\}_{l=1}^{L\ H}$，、$_{h=1}$在作用上类似于标准 SVGP 中的诱导位置：它们概括了训练集，但仅关注不确定性行为。因此，根据核度量，对于与全局键相似度较低的查询，公式 3.15 中的后验方差仍会增加。通过在每一层传播不确定性，对于与训练数据截然不同的输入序列，我们仍能获得增大的不确定性，从而用户仍可获知“模型何时并不了解”（Gal，2016年）。

B.7 表格中的结果

我们在表格中展示所有实验的数值结果（均值±标准误）。

我们在表格中展示分布内结果 B.4至B.10。

我们在表格中展示分布外鲁棒性结果B.11 至 B.27。

我们在表格中展示 OOD 检测结果B.28至 B.32。

表 B.4：分布内性能：基于 IMDB 的情感分析

	模型	准确率	NLL	ECE	MCE
SDP	MLE	0.8806±0.0016	0.3267±0.0068	0.0548±0.0029	0.1432±0.0093
核	MLE	0.8822±0.0019	0.3202±0.0136	0.0519±0.0055	0.1568±0.0273
	TS	0.8822±0.0019	0.3202±0.0136	0.0519±0.0055	0.1568±0.0004
	MFVI	0.8799±0.0022	0.3476±0.0230	0.0585±0.0106	0.1493±0.0271
	MCD	0.8817±0.0014	0.2986±0.0070	0.0285±0.0058	0.0910±0.0218
	KFLLLA	0.8822±0.0019	0.4366±0.0015	0.1905±0.0034	0.2263±0.0048
	SNGP	0.8783±0.0013	0.3499±0.0080	0.0646±0.0037	0.1741±0.0106
	SGPA	0.8875±0.0004	0.2823±0.0027	0.0215±0.0034	0.0647±0.0105

表 B.5：分布内性能：使用 CoLA 的语言可接受性

	模型	MCC	NLL	ECE	MCE
SDP	MLE	25.0408±0.8223	1.8976±0.0223	0.2654±0.0039	0.3887±0.0230
核	MLE	26.0722±0.4300	2.0083±0.0432	0.2643±0.0077	0.3702±0.0217
	MFVI	21.0864±1.8124	1.0809±0.0092	0.2018±0.0080	0.266±0.0113
	MCD	26.4399±0.4939	0.9121±0.0154	0.2113±0.0049	0.3233±0.0066
	KFLLLA	26.0295±0.4050	0.6023±0.0077	0.0241±0.0044	0.1063±0.0204
	SNGP	27.8715±1.2584	1.2075±0.0418	0.2325±0.0067	0.3230±0.0146
	SGPA	27.2380±1.1188	0.9070±0.0207	0.2076±0.0082	0.3062±0.0097

表 B.6：分布内性能：CIFAR10 无数据增强图像分类

	模型	准确率	NLL	ECE	MCE
SDP	MLE	0.7439±0.0015	2.2940±0.0277	0.2703±0.0021	0.4509±0.0079
核	MLE	0.7811±0.0019	1.3395±0.0135	0.2082±0.0010	0.3724±0.0103
	TS	0.7811±0.0019	0.8254±0.0077	0.1357±0.0012	0.2499±0.0081
	MFVI	0.7202±0.0024	0.7995±0.0073	0.0331±0.0023	0.0664±0.0064
	MCD	0.7910±0.0015	0.7789±0.0063	0.0799±0.0012	0.1501±0.0055
	KFLLLA	0.7752±0.0019	0.7201±0.0242	0.0710±0.0067	0.1497±0.0217
	SNGP	0.7837±0.0025	0.7534±0.0031	0.0745±0.0003	0.1443±0.0049
	SGPA	0.7787±0.0024	0.6968±0.0032	0.0589±0.0012	0.0825±0.0066
	DE	0.8235	0.6934	0.0504	0.0699
	SGPAE	0.8172	0.5657	0.0184	0.0579

表 B.7：分布内性能：使用数据增强的 CIFAR10 图像分类

	模型	准确率	NLL	ECE	MCE
SDP	MLE	0.8898±0.0009	0.6073±0.0330	0.1526±0.0043	0.3250±0.0118
核	MLE	0.8442±0.0057	0.4700±0.0127	0.0408±0.0031	0.0951±0.0116
	TS	0.8442±0.0057	0.4552±0.0176	0.0326±0.0028	0.0532±0.0077
	MFVI	0.6967±0.0022	0.8643±0.0078	0.0428±0.0072	0.0635±0.0083
	MCD	0.8437±0.0074	0.4541±0.0190	0.0122±0.0013	0.0428±0.0049
	KFLLLA	0.8445±0.0057	0.4554±0.0158	0.0279±0.0022	0.0570±0.0090
	SNGP	0.8286±0.0118	0.4912±0.0351	0.0076±0.0007	0.0398±0.0051
	SGPA	0.8475±0.0048	0.4489±0.0121	0.0086±0.0013	0.0395±0.0037
	DE	0.8654	0.4062	0.0079	0.0724
	SGPAE	0.8691	0.3885	0.0068	0.0421

表 B.8：分布内性能：未使用数据增强的 CIFAR100 图像分类

	模型	准确率	NLL	ECE	MCE
SDP	MLE	0.4893±0.0022	5.7677±0.0492	0.5276±0.0021	0.6216±0.0076
核	MLE	0.5216±0.0100	4.3898±0.0829	0.4061±0.0023	0.5696±0.0093
	TS	0.5216±0.0100	2.5863±0.0527	0.2769±0.0013	0.4067±0.0032
	MFVI	0.4117±0.0015	2.4195±0.0113	0.0753±0.0002	0.1021±0.0057
	MCD	0.5341±0.0096	2.7122±0.0329	0.1791±0.0013	0.2651±0.0116
	KFLLLA	0.5084±0.0103	3.2984±0.0210	0.1682±0.0054	0.7690±0.0159
	SNGP	0.4715±0.0039	2.8296±0.0338	0.2451±0.0005	0.3028±0.0082
	SGPA	0.5302±0.0071	2.5643±0.0813	0.1730±0.0013	0.2463±0.0071
	DE	0.5815	2.4887	0.1026	0.1381
	SGPAE	0.5700	1.9434	0.0467	0.0704

表 B.9：分布内性能：使用数据增强的 CIFAR100 图像分类

	模型	准确率	NLL	ECE	MCE
SDP	MLE	0.5984±0.0024	1.8743±0.2032	0.1879±0.0006	0.2792±0.0512
核	MLE	0.6053±0.0048	1.5254±0.0195	0.1066±0.0036	0.1578±0.0159
	TS	0.6053±0.0048	1.4512±0.0251	0.0303±0.0027	0.0825±0.0109
	MFVI	0.4496±0.0045	2.1132±0.0105	0.0527±0.0063	0.1339±0.0156
	MCD	0.6149±0.0074	1.4255±0.0275	0.0231±0.0032	0.0468±0.0056
	KFLLLA	0.5994±0.0038	1.4661±0.0206	0.0198±0.0022	0.0474±0.0071
	SNGP	0.5645±0.0106	1.5839±0.0416	0.0254±0.0018	0.0424±0.0060
	SGPA	0.6154±0.0010	1.4486±0.0055	0.0364±0.0055	0.0612±0.0067
	DE	0.6471	1.3023	0.0565	0.0986
	SGPEA	0.65	1.2761	0.0191	0.0593

表 B.10：分布内性能：使用 ZINC 数据集进行图属性回归

	模型	MAE	RMSE	NLL
SDP	MLE	0.5733±0.0084	1.1459±0.0139	1.1459±0.0139
核	MLE	0.5191±0.0344	1.0977±0.0869	1.0977±0.0869
	MCD	0.4385±0.0163	0.8369±0.0308	0.9264±0.0317
	MFVI	0.5702±0.0086	1.1050±0.0115	1.1683±0.0101
	SGPA	0.4341±0.0142	0.8199±0.0195	0.9319±0.0242
	DE	0.4711	0.8635	1.0949
	SGPAE	0.4557	0.8479	1.0118

表 B.11：分布外鲁棒性：CoLA数据集上的语言可接受性

	模型	MCC	NLL	ECE	MCE
SDP	MLE	19.4835±2.0234	2.0574±0.0516	0.2914±0.0098	0.4913±0.0418
核	MLE	18.6467±1.6116	2.3565±0.0974	0.2909±0.0067	0.3864±0.0303
	MFVI	16.6363±1.5793	1.1349±0.0445	0.2184±0.0035	0.3289±0.0151
	MCD	20.4774±1.4727	1.0114±0.0234	0.2370±0.0075	0.3439±0.0177
	KFLLLA	18.6250±1.6321	0.6299±0.0067	0.0361±0.0065	0.0876±0.0065
	SNGP	23.6410±1.5986	1.3070±0.0412	0.2537±0.0049	0.3828±0.0189
	SGPA	22.9190±1.5986	0.9514±0.0261	0.2257±0.0069	0.3399±0.0294

表 B.12：未使用数据增强、在干净数据上训练的视觉变换器在 CIFAR10-C 上的准确率。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.7120±0.0011	0.6704±0.0011	0.6359±0.0014	0.5883±0.0017	0.5354±0.0015
核	MLE	0.7254±0.0019	0.6632±0.0028	0.6139±0.0023	0.5494±0.0028	0.4863±0.0027
	MFVI	0.6606±0.0093	0.6109±0.0083	0.5730±0.0077	0.5266±0.0073	0.4778±0.0065
	MCD	0.7327±0.0013	0.6717±0.0015	0.6229±0.0016	0.5603±0.0019	0.4982±0.0020
	KFLLLA	0.7186±0.0014	0.6580±0.0035	0.6092±0.0038	0.5462±0.0042	0.4825±0.0038
	SNGP	0.7281±0.0024	0.6676±0.0025	0.6208±0.0020	0.5622±0.0020	0.5006±0.0019
	SGPA	0.7175±0.0038	0.6559±0.0042	0.6067±0.0040	0.5466±0.0033	0.4832±0.0046
	DE	0.7700	0.7089	0.6585	0.5908	0.5246
	SGPAE	0.7516	0.6911	0.6401	0.5779	0.5173

表 B.13：在干净数据上使用数据增强训练的视觉变换器在 CIFAR10-C 上的准确率。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.8516±0.0010	0.8139±0.0011	0.7752±0.0010	0.7165±0.0013	0.6500±0.0012
核	MLE	0.7944±0.0061	0.7499±0.0074	0.7090±0.0082	0.6465±0.0085	0.5807±0.0094
	MFVI	0.6605±0.0016	0.6222±0.0024	0.5875±0.0028	0.5377±0.0032	0.4913±0.0040
	MCD	0.7939±0.0068	0.7480±0.0082	0.7052±0.0089	0.6415±0.0092	0.5751±0.0097
	KFLLLA	0.7949±0.0030	0.7508±0.0037	0.7100±0.0041	0.6480±0.0043	0.5821±0.0047
	SNGP	0.7783±0.0062	0.7336±0.0069	0.6932±0.0069	0.6314±0.0062	0.5710±0.0057
	SGPA	0.7910±0.0027	0.7426±0.0021	0.6946±0.0075	0.6356±0.0033	0.5712±0.0020
	DE	0.8237	0.7819	0.7428	0.6795	0.6130
	SGPAE	0.8105	0.7646	0.7220	0.6581	0.5914

表 B.14：未使用数据增强、在干净数据上训练的视觉变换器在 CIFAR100-C 上的准确率。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.3947±0.0022	0.3484±0.0018	0.3223±0.0014	0.2805±0.0013	0.2402±0.0009
核	MLE	0.4456±0.0075	0.3769±0.0058	0.3373±0.0049	0.2835±0.0034	0.2339±0.0029
	MFVI	0.3380±0.0020	0.2832±0.0022	0.2540±0.0018	0.2155±0.0014	0.1803±0.0015
	MCD	0.4526±0.0074	0.3840±0.0061	0.3446±0.0052	0.2913±0.0037	0.2403±0.0031
	KFLLLA	0.4350±0.0070	0.3656±0.0053	0.3251±0.0045	0.2717±0.0030	0.2229±0.0025
	SNGP	0.4053±0.0020	0.3455±0.0014	0.3093±0.0015	0.2624±0.0015	0.2203±0.0011
	SGPA	0.4472±0.0050	0.3764±0.0035	0.3371±0.0027	0.2828±0.0021	0.2317±0.0022
	DE	0.5054	0.4289	0.3858	0.3234	0.2690
	SGPAE	0.4848	0.4104	0.3675	0.3087	0.2539

表 B.15：在干净数据上使用数据增强训练的视觉变换器的 CIFAR100-C 准确率。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.5384±0.0030	0.4854±0.0038	0.4464±0.0041	0.3930±0.0037	0.3410±0.0029
核	MLE	0.5327±0.0018	0.4700±0.0018	0.4264±0.0017	0.3690±0.0017	0.3134±0.0013
	MFVI	0.3991±0.0019	0.3535±0.0017	0.3224±0.0019	0.2836±0.0017	0.2444±0.0014
	MCD	0.5397±0.0020	0.4775±0.0018	0.4349±0.0015	0.3777±0.0011	0.3213±0.0010
	KFLLLA	0.5196±0.0019	0.4577±0.0018	0.4148±0.0017	0.3581±0.0014	0.3035±0.0014
	SNGP	0.4979±0.0038	0.4440±0.0035	0.4042±0.0034	0.3534±0.0029	0.3066±0.0023
	SGPA	0.5189±0.0054	0.4603±0.0010	0.4181±0.0013	0.3625±0.0014	0.3061±0.0014
	DE	0.5742	0.5104	0.4649	0.4039	0.3449
	SGPAE	0.5592	0.4936	0.4476	0.3908	0.3316

表 B.16：在未使用数据增强的情况下以干净数据训练的视觉变换器在 CIFAR10-C 上的负对数似然。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	2.6411±0.0202	3.1620±0.0202	3.6449±0.0231	4.3843±0.0283	5.2626±0.0287
核	MLE	1.7906±0.0179	2.3460±0.0321	2.8899±0.0385	3.6899±0.0544	4.5003±0.0811
	MFVI	0.9761±0.0274	1.1345±0.0267	1.2737±0.0259	1.4705±0.0237	1.7038±0.0222
	MCD	1.1466±0.0099	1.5153±0.0168	1.8926±0.0234	2.4692±0.0315	3.1086±0.0431
	KFLLLA	0.9573±0.0498	1.2092±0.0577	1.4435±0.0683	1.7889±0.0937	2.1537±0.1198
	SNGP	1.0579±0.0066	1.3681±0.0115	1.6570±0.0129	2.0807±0.0136	2.5271±0.0146
	SGPA	0.9603±0.0128	1.2340±0.0213	1.4972±0.0283	1.8951±0.0383	2.3812±0.0586
	DE	0.9317	1.2424	1.5716	2.0911	2.6373
	SGPAE	0.7816	0.9985	1.2125	1.5365	1.9349

表 B.17：使用数据增强训练的视觉变换器在 CIFAR10-C 上的负对数似然。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.8306±0.0422	1.0712±0.0559	1.3665±0.0753	1.8795±0.1085	2.5810±0.1571
核	MLE	0.6364±0.0140	0.7960±0.0207	0.9614±0.0296	1.2404±0.0416	1.5818±0.0538
	MFVI	0.9616±0.0044	1.0706±0.0061	1.1802±0.0098	1.3612±0.0160	1.5629±0.0248
	MCD	0.6067±0.0180	0.7579±0.0236	0.9144±0.0304	1.1707±0.0363	1.4977±0.0420
	KFLLLA	0.6104±0.0080	0.7565±0.0108	0.9044±0.0144	1.1496±0.0183	1.4487±0.0230
	SNGP	0.6439±0.0187	0.7874±0.0218	0.9304±0.0225	1.1773±0.0223	1.4671±0.0225
	SGPA	0.6204±0.0054	0.7739±0.0054	0.9298±0.0047	1.1873±0.0118	1.5058±0.0226
	DE	0.5641	0.7028	0.8441	1.0737	1.3595
	SGPAE	0.5230	0.6503	0.7814	1.0027	1.2835

表 B.18：未使用数据增强、在干净数据上训练的视觉变换器在 CIFAR100-C 上的负对数似然。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	7.4649±0.0475	8.5233±0.0524	9.1964±0.0527	10.4061±0.0555	11.8540±0.0649
核	MLE	5.5126±0.0578	6.7077±0.0416	7.5503±0.0237	9.0093±0.0353	10.5647±0.0699
	MFVI	3.0138±0.0132	3.4762±0.0136	3.7921±0.0135	4.3295±0.0153	4.9042±0.0200
	MCD	3.8996±0.0266	4.8412±0.0178	5.5397±0.0269	6.7984±0.0591	8.2454±0.0959
	KFLLLA	3.4366±0.0172	3.5660±0.0160	3.6457±0.0152	3.7645±0.0135	3.8862±0.0115
	SNGP	3.6404±0.0297	4.3067±0.0360	4.7994±0.0446	5.6242±0.0502	6.5029±0.0504
	SGPA	3.6565±0.1290	4.5188±0.1401	5.1522±0.1510	6.3096±0.1784	7.6306±0.2161
	DE	3.1930	4.0004	4.6072	5.7357	6.9765
	SGPAE	2.8407	3.5393	4.0466	4.9890	6.0803

表 B.19：在带数据增强的干净数据上训练的视觉变换器在 CIFAR100-C 上的负对数似然。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	2.2595±0.2276	2.6313±0.2735	2.9786±0.3148	3.5393±0.3720	4.1865±0.4426
核	MLE	1.9695±0.0167	2.3721±0.0260	2.7423±0.0346	3.3355±0.0500	3.9813±0.0602
	MFVI	2.3501±0.0073	2.5702±0.0084	2.7571±0.0112	3.0582±0.0123	3.3805±0.0147
	MCD	1.8337±0.0048	2.1874±0.0073	2.5076±0.0112	3.0263±0.0205	3.6196±0.0273
	KFLLLA	1.8645±0.0092	2.2008±0.0095	2.4935±0.0095	2.9465±0.0112	3.4418±0.0135
	SNGP	1.9167±0.0161	2.2067±0.0162	2.4753±0.0163	2.9166±0.0133	3.3598±0.0116
	SGPA	1.9307±0.0075	2.3154±0.0104	2.6598±0.0130	3.2105±0.0151	3.8575±0.0180
	DE	1.7354	2.0787	2.3857	2.8802	3.4604
	SGPAE	1.6583	1.9839	2.2802	2.7592	3.2976

表 B.20：在干净数据上训练且未使用数据增强的视觉变换器的 CIFAR10-C 期望校准误差。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.2919±0.0021	0.3297±0.0021	0.3623±0.0023	0.3964±0.0026	0.4458±0.0030
核	MLE	0.2514±0.0019	0.3003±0.0020	0.3396±0.0026	0.3898±0.0028	0.4355±0.0032
	MFVI	0.0324±0.0013	0.0767±0.0019	0.1031±0.0021	0.1273±0.0025	0.1684±0.0025
	MCD	0.1191±0.0011	0.1589±0.0013	0.1928±0.0013	0.2390±0.0015	0.2831±0.0017
	KFLLLA	0.1044±0.0012	0.1437±0.0017	0.1827±0.0017	0.2306±0.0029	0.2674±0.0035
	SNGP	0.1095±0.0011	0.1477±0.0013	0.1860±0.0015	0.2337±0.0015	0.2724±0.0018
	SGPA	0.0894±0.0013	0.1305±0.0018	0.1653±0.0020	0.2137±0.0024	0.2562±0.0024
	DE	0.0771	0.1095	0.1406	0.1826	0.2168
	SGPAE	0.0323	0.0644	0.0946	0.1381	0.1736

表 B.21：在干净数据上使用数据增强训练的视觉变换器的 CIFAR10-C 期望校准误差。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.1731±0.0015	0.1962±0.0015	0.2094±0.0017	0.2582±0.0021	0.3046±0.0031
核	MLE	0.0666±0.0014	0.0920±0.0016	0.1170±0.0018	0.1605±0.0027	0.2090±0.0036
	MFVI	0.0295±0.0006	0.0519±0.0008	0.0715±0.0015	0.1097±0.0019	0.1534±0.0026
	MCD	0.0245±0.0006	0.0433±0.0008	0.0657±0.0013	0.1084±0.0017	0.1545±0.0023
	KFAC-LLLA	0.0352±0.0006	0.0587±0.0008	0.0797±0.0014	0.1175±0.0018	0.1634±0.0024
	SNGP	0.0260±0.0004	0.0368±0.0009	0.0555±0.0016	0.0929±0.0025	0.1370±0.0035
	SGPA	0.0217±0.0004	0.0368±0.0009	0.0565±0.0017	0.0971±0.0027	0.1421±0.0038
	DE	0.0291	0.0369	0.0549	0.0904	0.1338
	SGPAE	0.0224	0.0351	0.0498	0.0821	0.1196

表 B.22：在干净数据上训练且未使用数据增强的视觉变换器在 CIFAR100-C 上的期望校准误差。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.5735±0.0032	0.6016±0.0028	0.6272±0.0031	0.6450±0.0027	0.6799±0.0024
核	MLE	0.4588±0.0028	0.5095±0.0030	0.5393±0.0028	0.5826±0.0019	0.6209±0.0020
	MFVI	0.1014±0.0012	0.1458±0.0011	0.1592±0.0012	0.2092±0.0013	0.2575±0.0023
	MCD	0.2376±0.0010	0.2799±0.0009	0.3086±0.0017	0.3562±0.0023	0.4012±0.0031
	KFAC-LLLA	0.1753±0.0029	0.1740±0.0031	0.1553±0.0029	0.1478±0.0020	0.1427±0.0020
	SNGP	0.2427±0.0010	0.2839±0.0010	0.3236±0.0017	0.3621±0.0023	0.4056±0.0031
	SGPA	0.2337±0.0012	0.2763±0.0011	0.3045±0.0012	0.3520±0.0012	0.3963±0.0023
	DE	0.1696	0.2117	0.2400	0.2877	0.3283
	SGPAE	0.0972	0.1390	0.1665	0.2149	0.2605

表 B.23：使用数据增强在干净数据上训练的视觉变换器的 CIFAR100-C 期望校准误差。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.2179±0.0022	0.2485±0.0020	0.2901±0.0023	0.3327±0.0025	0.3816±0.0025
核	MLE	0.1459±0.0020	0.1793±0.0021	0.2067±0.0022	0.2454±0.0023	0.2797±0.0023
	MFVI	0.0478±0.0032	0.0409±0.0038	0.0510±0.0040	0.0681±0.0042	0.0824±0.0044
	MCD	0.0614±0.0021	0.0919±0.0025	0.1197±0.0028	0.1596±0.0028	0.1979±0.0032
	KFAC-LLLA	0.0493±0.0045	0.0635±0.0057	0.0862±0.0053	0.1323±0.0045	0.1764±0.0047
	SNGP	0.0368±0.0019	0.0543±0.0025	0.0779±0.0027	0.1192±0.0025	0.1429±0.0031
	SGPA	0.0742±0.0040	0.1060±0.0045	0.1341±0.0046	0.1733±0.0044	0.2124±0.0045
	DE	0.0355	0.0562	0.0780	0.1144	0.1440
	SGPAE	0.0434	0.0511	0.0680	0.1002	0.1308

表 B.24：在未使用数据增强的情况下，使用干净数据训练的视觉变换器在 CIFAR10-C 上的最大校准误差。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.4666±0.0027	0.4840±0.0026	0.5041±0.0018	0.5280±0.0019	0.5534±0.0020
核	MLE	0.4038±0.0028	0.4362±0.0020	0.4655±0.0020	0.5051±0.0033	0.5442±0.0036
	MFVI	0.0598±0.0112	0.0852±0.0150	0.1125±0.0147	0.1586±0.0147	0.2011±0.0149
	MCD	0.1902±0.0021	0.2402±0.0025	0.2807±0.0026	0.3413±0.0036	0.3949±0.0041
	KFLLLA	0.1987±0.0251	0.2362±0.0241	0.2729±0.0238	0.3248±0.0220	0.3733±0.0210
	SNGP	0.1854±0.0018	0.2337±0.0021	0.2710±0.0011	0.3229±0.0021	0.3798±0.0008
	SGPA	0.1273±0.0042	0.1801±0.0046	0.2212±0.0061	0.2807±0.0057	0.3340±0.0047
	DE	0.1051	0.1402	0.1813	0.2476	0.3091
	SGPAE	0.0570	0.0816	0.1173	0.1757	0.2208

表 B.25：在干净数据上使用数据增强训练的视觉变换器在 CIFAR10-C 上的最大校准误差。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.3500±0.0125	0.3649±0.0126	0.3870±0.0123	0.4186±0.0122	0.4561±0.0129
核	MLE	0.1188±0.0093	0.1451±0.0091	0.1776±0.0090	0.2341±0.0092	0.2886±0.0090
	MFVI	0.0772±0.0048	0.0965±0.0025	0.1283±0.0052	0.1903±0.0067	0.2261±0.0090
	MCD	0.0566±0.0032	0.0788±0.0031	0.1110±0.0048	0.1720±0.0062	0.2298±0.0057
	KFLLLA	0.0808±0.0031	0.1035±0.0028	0.1328±0.0030	0.1845±0.0036	0.2396±0.0036
	SNGP	0.0557±0.0013	0.0797±0.0029	0.1093±0.0039	0.1764±0.0040	0.2318±0.0044
	SGPA	0.0569±0.0022	0.0781±0.0026	0.1059±0.0030	0.1601±0.0054	0.2167±0.0050
	DE	0.0868	0.0855	0.1003	0.1334	0.1705
	SGPAE	0.0627	0.0659	0.0882	0.1262	0.1776

表 B.26：在未进行数据增强的干净数据上训练的视觉变换器的 CIFAR100-C 最大校准误差。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.6608±0.0014	0.6852±0.0014	0.7009±0.0022	0.7251±0.0020	0.7542±0.0011
核	MLE	0.6037±0.0032	0.6428±0.0031	0.6649±0.0027	0.7037±0.0007	0.7385±0.0010
	MFVI	0.1925±0.0031	0.2594±0.0039	0.3069±0.0055	0.3809±0.0044	0.4562±0.0027
	MCD	0.3424±0.0042	0.4103±0.0047	0.4554±0.0045	0.5255±0.0052	0.5902±0.0038
	KFLLLA	0.7691±0.0165	0.7632±0.0181	0.7512±0.0194	0.7248±0.0146	0.6699±0.0228
	SNGP	0.3631±0.0042	0.4201±0.0044	0.4624±0.0039	0.5177±0.0043	0.5749±0.0037
	SGPA	0.3217±0.0057	0.3886±0.0083	0.4353±0.0093	0.5047±0.0083	0.5726±0.0072
	DE	0.1767	0.2257	0.2655	0.3522	0.4202
	SGPAE	0.1247	0.1766	0.2209	0.3142	0.3965

表 B.27：使用数据增强在干净数据上训练的视觉变换器的 CIFAR100-C 最大校准误差。

		偏斜强度				
	模型	1	2	3	4	5
SDP	MLE	0.3142±0.0470	0.3487±0.0458	0.3816±0.0451	0.4308±0.0412	0.4839±0.0387
核	MLE	0.2237±0.0136	0.2650±0.0122	0.3098±0.0131	0.3733±0.0117	0.4381±0.0140
	MFVI	0.1330±0.0083	0.1173±0.0064	0.1275±0.0051	0.1751±0.0056	0.2208±0.0069
	MCD	0.0995±0.0066	0.1418±0.0067	0.1845±0.0067	0.2568±0.0081	0.3240±0.0082
	KFLLLA	0.0893±0.0040	0.1278±0.0048	0.1689±0.0057	0.2355±0.0067	0.2999±0.0076
	SNGP	0.0636±0.0012	0.0972±0.0013	0.1377±0.0015	0.2007±0.0017	0.2620±0.0012
	SGPA	0.1120±0.0022	0.1582±0.0034	0.2062±0.0040	0.2776±0.0028	0.3455±0.0030
	DE	0.0920	0.0971	0.1220	0.1906	0.2444
	SGPAE	0.0722	0.0888	0.1088	0.1766	0.2291

表 B.28：使用未经数据增强的 CIFAR10 训练的 ViT 进行分布外检测的 AUROC 和 AUPR 指标。

		分布外：CIFAR100		分布外：SVHN		分布外：MINIIMAGENET	
	模型	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
SDP	MLE	0.6893±0.0018	0.6433±0.0023	0.6983±0.0109	0.8206±0.0074	0.7115±0.0018	0.9219±0.0006
核	MLE	0.7208±0.0029	0.6774±0.0029	0.7477±0.0118	0.8518±0.0068	0.7325±0.0016	0.9300±0.0004
	MFVI	0.7177±0.0055	0.6722±0.0050	0.7059±0.0195	0.7944±0.0134	0.7531±0.0036	0.9407±0.0008
	MCD	0.7426±0.0016	0.7046±0.0016	0.6860±0.0138	0.8045±0.0083	0.7443±0.0019	0.9356±0.0006
	KFLLLA	0.7228±0.0056	0.6889±0.0055	0.7599±0.0105	0.8345±0.0069	0.7855±0.0046	0.9487±0.0015
	SNGP	0.7391±0.0016	0.7023±0.0029	0.7135±0.0125	0.8258±0.0083	0.7391±0.0050	0.9341±0.0015
	SGPA	0.7498±0.0010	0.7111±0.0023	0.7198±0.0085	0.8125±0.0067	0.7501±0.0013	0.9382±0.0004
	DE	0.7757±0.0000	0.7423±0.0000	0.7819±0.0000	0.8512±0.0000	0.8007±0.0000	0.9520±0.0000
	SGPAE	0.7872±0.0000	0.7482±0.0000	0.7600±0.0000	0.8619±0.0000	0.8031±0.0000	0.9529±0.0000

表 B.29：使用数据增强在 CIFAR10 上训练的视觉变换器进行分布外检测的 AUROC 和 AUPR 指标。

		分布外：CIFAR100		分布外：SVHN		分布外：MINIIMAGENET	
	模型	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
SDP	MLE	0.8245±0.0007	0.7780±0.0013	0.8738±0.0020	0.9308±0.0015	0.8375±0.0009	0.9585±0.0005
核	MLE	0.7908±0.0034	0.7523±0.0028	0.8748±0.0023	0.9338±0.0015	0.8148±0.0027	0.9555±0.0005
	MFVI	0.7009±0.0043	0.6530±0.0045	0.7652±0.0179	0.8468±0.0109	0.7506±0.0073	0.9379±0.0027
	MCD	0.7995±0.0038	0.7614±0.0030	0.8754±0.0040	0.9304±0.0029	0.8282±0.0029	0.9595±0.0007
	KFLLLA	0.7989±0.0041	0.7617±0.0036	0.8884±0.0036	0.9415±0.0022	0.8273±0.0034	0.9591±0.0007
	SNGP	0.7889±0.0092	0.7543±0.0104	0.8776±0.0044	0.9326±0.0032	0.8125±0.0072	0.9551±0.0020
	SGPA	0.8007±0.0014	0.7630±0.0013	0.8746±0.0061	0.9281±0.0043	0.8319±0.0014	0.9600±0.0004
	DE	0.8230±0.0000	0.7869±0.0000	0.9195±0.0000	0.9575±0.0000	0.8555±0.0000	0.9666±0.0000
	SGPAE	0.8264±0.0000	0.7917±0.0000	0.9004±0.0000	0.9452±0.0000	0.8606±0.0000	0.9677±0.0000

表 B.30：使用未经数据增强训练的 CIFAR100 视觉变换器进行分布外检测的 AUROC 和 AUPR 指标。

		分布外：CIFAR10		分布外：SVHN		分布外：MINIIMAGENET	
	模型	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
SDP	MLE	0.6091±0.0023	0.5756±0.0030	0.6242±0.0014	0.7856±0.0015	0.6512±0.0018	0.9042±0.0007
核	MLE	0.6355±0.0052	0.5983±0.0045	0.6725±0.0111	0.8171±0.0062	0.6582±0.0035	0.9071±0.0010
	MFVI	0.6277±0.0036	0.5900±0.0037	0.6242±0.0108	0.7729±0.0079	0.6496±0.0054	0.9051±0.0024
	MCD	0.6729±0.0069	0.6310±0.0076	0.6054±0.0057	0.7597±0.0038	0.6859±0.0014	0.9155±0.0005
	KFLLLA	0.6297±0.0059	0.5920±0.0071	0.6733±0.0159	0.8101±0.0086	0.6915±0.0057	0.9194±0.0017
	SNGP	0.6448±0.0016	0.6080±0.0020	0.6764±0.0096	0.8093±0.0063	0.6823±0.0023	0.9161±0.0009
	SGPA	0.6712±0.0052	0.6297±0.0057	0.6454±0.0147	0.7854±0.0099	0.6911±0.0028	0.9170±0.0009
	DE	0.6848±0.0000	0.6377±0.0000	0.7388±0.0000	0.8425±0.0000	0.7394±0.0000	0.9300±0.0000
	SGPAE	0.6925±0.0000	0.6461±0.0000	0.6961±0.0000	0.8213±0.0000	0.7398±0.0000	0.9304±0.0000

表 B.31：使用数据增强在 CIFAR100 上训练的视觉变换器进行分布外检测的 AUROC 和 AUPR 指标。

		分布外：CIFAR10		分布外：SVHN		分布外：MINIIMAGENET	
	模型	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
SDP	MLE	0.6778±0.0022	0.6312±0.0001	0.7652±0.0093	0.8769±0.0055	0.7430±0.0065	0.9346±0.0020
核	MLE	0.6899±0.0026	0.6486±0.0023	0.7570±0.0056	0.8670±0.0025	0.7536±0.0038	0.9374±0.0012
	MFVI	0.6403±0.0042	0.5924±0.0035	0.7485±0.0106	0.8640±0.0068	0.7269±0.0099	0.9317±0.0030
	MCD	0.7047±0.0025	0.6568±0.0029	0.7972±0.0040	0.8885±0.0023	0.7780±0.0023	0.9440±0.0008
	KFLLLA	0.6924±0.0011	0.6510±0.0022	0.8164±0.0043	0.9104±0.0053	0.7726±0.0084	0.9435±0.0024
	SNGP	0.6813±0.0034	0.6388±0.0034	0.7616±0.0090	0.8701±0.0052	0.7486±0.0048	0.9367±0.0014
	SGPA	0.7056±0.0012	0.6586±0.0011	0.8120±0.0035	0.8971±0.0027	0.7755±0.0012	0.9430±0.0004
	DE	0.7171±0.0000	0.6718±0.0000	0.8459±0.0000	0.9173±0.0000	0.8076±0.0000	0.9517±0.0000
	SGPAE	0.7198±0.0000	0.6719±0.0000	0.8686±0.0000	0.9298±0.0000	0.8152±0.0000	0.9536±0.0000

表 B.32：使用在 ZINC 上训练的 Transformer 进行分布外检测的 AUROC 和 AUPR 指标。

	模型	AUROC	AUPR
SDP	MCD	0.4566±0.0072	0.4889±0.0029
核	MFVI	0.7254±0.0632	0.6821±0.0512
	MCD	0.8797±0.0739	0.8629±0.0571
	SGPA	0.8968±0.0317	0.8705±0.0335
	DE	0.9783±0.0000	0.9697±0.0000
	SGPAE	0.9884±0.0000	0.9727±0.0000

附录 C

第 4 章补充材料

C.1 将先验诱导协方差 $\mathbf{K}_{\mathbf{uu}}^{(t)}$ 作为直接常微分方程演化来计算

尽管在正文的实验中，我们基于 RFF近似计算 $\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}$ ，但为求完整起见，这里提供另一种计算方法的详细推导。当诱导函数通过 HiPPO-LegS 定义时，该方法直接将 $\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}$ 作为矩阵常微分方程进行演化。回顾一下，

$$[\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}]_{\ell,m} = \iint k(x,x')\phi_\ell^{(t)}(x)\phi_m^{(t)}(x')\mathrm{d}x\mathrm{d}x', \tag{C.1}$$

其中， $\phi_\ell^{(t)}(x) = g_\ell^{(t)}(x)\omega^{(t)}(x)$ 是 HiPPO-LegS 框架下的时变基函数。

对$[\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}]_{\ell,\ m}$ 关于 t求导可得

$$\frac{d}{dt}[\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}]_{\ell,m} = \iint k(x,x')\frac{\partial}{\partial t}\left[\phi_\ell^{(t)}(x)\phi_m^{(t)}(x')\right]\mathrm{d}x\mathrm{d}x'. \tag{C.2}$$

应用乘积法则:

$$\frac{\mathrm{d}}{\mathrm{d}t}[\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}]_{\ell,m} = \iint k(x,x')\frac{\partial}{\partial t}\phi_\ell^{(t)}(x)\phi_m^{(t)}(x')\mathrm{d}x\mathrm{d}x' + \iint k(x,x')\phi_\ell^{(t)}(x)\frac{\partial}{\partial t}\phi_m^{(t)}(x')\mathrm{d}x\mathrm{d}x'. \tag{C.3}$$

在 HiPPO-LegS 中，每个 $\phi_\ell^{(t)}(x)$ 都遵循一个常微分方程，该方程由 $[0,t]$ 上低阶的缩放勒让德多项式以及 $x = t$ 处的狄拉克 δ 函数边界项控制，正如我们在

章节2.1.2。具体来说，

$$\frac{\partial}{\partial t}\phi_\ell^{(t)}(x) = -\frac{\sqrt{2\ell+1}}{t}\Big[\frac{\ell+1}{\sqrt{2\ell+1}}\phi_\ell^{(t)}(x) + \sqrt{2\ell-1}\phi_{\ell-1}^{(t)}(x) + \sqrt{2\ell-3}\phi_{\ell-2}^{(t)}(x)\cdots\Big] + \frac{1}{t}\,\delta_t(x), \tag{C.4}$$

其中 $\delta_t(x)$ 是位于 $x = t$ 处的狄拉克 δ 函数。

将此表达式代入积分后，可得到如下形式的边界项：$\int k(t,\ x')\ \phi_m^{(t)}(x')\ \mathrm{d}x'$，以及涉及低阶项 $\{[\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}]_{\ell,\ m},\ [\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}]_{\ell-1,\ m},\ \cdots\}$ 等。总结为矩阵形式可得

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{K}_{\boldsymbol{uu}}^{(t)} = \left[\boldsymbol{A}(t)\,\boldsymbol{K}_{\boldsymbol{uu}}^{(t)} + \boldsymbol{K}_{\boldsymbol{uu}}^{(t)}\boldsymbol{A}(t)^\top\right] + \frac{1}{t}\left[\tilde{\boldsymbol{B}}(t) + \tilde{\boldsymbol{B}}(t)^\top\right], \tag{C.5}$$

其中 lm 是 $\boldsymbol{K}_{\boldsymbol{uu}}^{(t)} \in \mathbb{R}^{M\times M}$ 的第 $[\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}]_{\ell,\ m}$，$\boldsymbol{A}(t) \in \mathbb{R}^{M\times M}$ 是式 2.14 中定义的 HiPPO-LegS 框架中的同一个下三角矩阵，并且 $\tilde{\boldsymbol{B}}(t) \in \mathbb{R}^{M\times M}$ 由边界贡献构造而成，如下所示

$$\tilde{\boldsymbol{B}}(t) = \boldsymbol{c}(t)\mathbf{1}_M, \tag{C.6}$$

其中 $\mathbf{1}_M \in \mathbb{R}^{1\times M}$ 是一个大小为 M 的全 1 行向量，$\boldsymbol{c}(t) \in \mathbb{R}^{M\times 1}$ 是系数向量，其每个元素为

$$[\boldsymbol{c}(t)]_\ell = \int k\,(t,x)\ \phi_\ell^{(t)}(x)\,\mathrm{d}x. \tag{C.7}$$

在 t 中离散化（例如采用欧拉格式）后，可以随时间递归更新 $\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}$ 和边界向量 $\boldsymbol{c}$（t）。

直接将 $\mathrm{K}_{\mathbf{uu}}^{(t)}$ 作为常微分方程演化的不稳定性。通过实验，我们发现，与 RFF 方法相比，直接常微分方程方法的稳定性较差。直观来看，这可以从两种演化形式的差异中看出，尤其是第一项。在 RFF 方法中，傅里叶特征演化的第一项形式为 $\boldsymbol{A}(t)\,\boldsymbol{Z}_w^{(t)}$，其中包含由算子 L_1 演化的向量：$\boldsymbol{X} \to \boldsymbol{A}(t)\boldsymbol{X}$。在直接常微分方程方法中，$\boldsymbol{K}_{\boldsymbol{uu}}^{(t)}$ 直接演化的第一项形式为 $\boldsymbol{A}(t)\,\boldsymbol{K}_{\boldsymbol{uu}}^{(t)} + \boldsymbol{K}_{\boldsymbol{uu}}^{(t)}\boldsymbol{A}(t)^\top$，这需要李雅普诺夫算子 L_2：$\boldsymbol{X} \to \boldsymbol{A}(t)\boldsymbol{X} + \boldsymbol{X}\,\boldsymbol{A}(t)^\top$。关键区别在于，$\mathrm{L}_2$ 的特征值为 $\{\lambda_i + \lambda_j\}$（其中$\lambda_i$ 和 λ_j 是 $\boldsymbol{A}(t)$ 的特征值）（Horn 和 Johnson，1991），而 L_1 的特征值为 $\{\lambda_i\}$。由于 HiPPO-LegS 使用对角线元素为负的下三角 $\boldsymbol{A}(t)$，其特征值全为负 $\lambda_i < 0$。因此，李雅普诺夫算子 L_2 的特征值约为 L_1，特征值的两倍负值，导致刚性常微分方程组的数值条件性较差。

C.2 F后验 OHSVGP 的有限基函数近似

这里，我们证明 $q(\boldsymbol{u}^{(t)})$ 是后验 OHSVGP $q_t(f)$ 的 HiPPO 系数分布。根据式 2.36，在任意索引 $\boldsymbol{X}$处评估的函数值的后验分布为 $q_t(\boldsymbol{f_X}) = \mathcal{N}(\boldsymbol{f_X}; \mathbf{K}_{\boldsymbol{f_X}\mathbf{u}}^{(t)}\mathbf{K}_{\mathbf{uu}}^{(t)-1}\boldsymbol{m}_{\mathbf{u}}^{(t)}, \mathbf{K}_{\boldsymbol{f_X}\boldsymbol{f_X}} - \mathbf{K}_{\boldsymbol{f_X}\mathbf{u}}^{(t)}\mathbf{K}_{\mathbf{uu}}^{(t)-1}[\mathbf{K}_{\mathbf{uu}}^{(t)} - \mathbf{S}_{\mathbf{u}}^{(t)}]\mathbf{K}_{\mathbf{uu}}^{(t)-1}\mathbf{K}^{(t)}{}_{\mathbf{u}\boldsymbol{f_X}})$。

基于此，我们按如下方式计算 $q_t(f)$ 的第 m 个 HiPPO 系数的均值，

$$
\begin{aligned}
& E_{q_t(f)}\left[\int f(x)\phi_m^{(t)}(x)dx\right] \\
&= \int E_{q_t(f)}\left[f(x)\right]\phi_m^{(t)}(x)dx \\
&= \left(\int \mathbf{K}_{f_x\mathbf{u}}^{(t)}\mathbf{K}_{\mathbf{uu}}^{(t)-1}\phi_m^{(t)}(x)dx\right)\boldsymbol{m}_{\mathbf{u}}^{(t)} \\
&= \left[\int\int k(x,x')\begin{pmatrix}\phi_1^{(t)}(x')\\ \cdots \\ \phi_M^{(t)}(x')\end{pmatrix}dx'\mathbf{K}_{\mathbf{uu}}^{(t)-1}\phi_m^{(t)}(x)dx\right]\boldsymbol{m}_{\mathbf{u}}^{(t)} \\
&= \left[\int\int k(x,x')\begin{pmatrix}\phi_1^{(t)}(x')\phi_m^{(t)}(x)\\ \cdots \\ \phi_M^{(t)}(x')\phi_m^{(t)}(x)\end{pmatrix}dx'dx\right]\mathbf{K}_{\mathbf{uu}}^{(t)-1}\boldsymbol{m}_{\mathbf{u}}^{(t)} \\
&= \left[\mathbf{K}_{\mathbf{uu}}^{(t)}\right]_{m,:}\mathbf{K}_{\mathbf{uu}}^{(t)-1}\boldsymbol{m}_{\mathbf{u}}^{(t)} \\
&= \left[\boldsymbol{m}_{\mathbf{u}}^{(t)}\right]_m,
\end{aligned}
\tag{C.8}
$$

这恰好是 $q(u_m^{(t)})$ 的变分均值。类似地， $q_t(f)$ 的第 l 个与第 m 个 HiPPO 系数之间的协方差可计算为

$$
\begin{aligned}
& E_{q_t(f)}\left[\left(\int f(x)\phi_l^{(t)}(x)dx\right)\left(\int f(x')\phi_m^{(t)}(x')dx'\right)\right] - \left[\boldsymbol{m}_{\mathbf{u}}^{(t)}\right]_l\left[\boldsymbol{m}_{\mathbf{u}}^{(t)}\right]_m \\
&= \int\int E_{q_t(f)}\left[f(x)f(x')\right]\phi_l^{(t)}(x)\phi_m^{(t)}(x')dxdx' - \left[\boldsymbol{m}_{\mathbf{u}}^{(t)}\right]_l\left[\boldsymbol{m}_{\mathbf{u}}^{(t)}\right]_m \\
&= \int\int \left(k(x,x') - \mathbf{K}_{f_x\mathbf{u}}^{(t)}\mathbf{K}_{\mathbf{uu}}^{(t)-1}[\mathbf{K}_{\mathbf{uu}}^{(t)} - \mathbf{S}_{\mathbf{u}}^{(t)}]\mathbf{K}_{\mathbf{uu}}^{(t)-1}\mathbf{K}_{\mathbf{u}f_{x'}}^{(t)}\right)\phi_l^{(t)}(x)\phi_m^{(t)}(x')dxdx' \\
&\qquad + \int\int E_{q_t(f)}\left[f(x)\right]\cancel{E_{q_t(f)}\left[f(x')\right]}\phi_l^{(t)}(x)\phi_m^{(t)}(x')dxdx' - \cancel{\left[\boldsymbol{m}_{\mathbf{u}}^{(t)}\right]_l\left[\boldsymbol{m}_{\mathbf{u}}^{(t)}\right]_m} \\
&= \left[\mathbf{K}_{\mathbf{uu}}^{(t)}\right]_{lm} \cancel{- \left(\int \mathbf{K}_{f_x\mathbf{u}}^{(t)}\mathbf{K}_{\mathbf{uu}}^{(t)-1}\phi_l^{(t)}(x)dx\right)\left[\mathbf{K}_{\mathbf{uu}}^{(t)} - \mathbf{S}_{\mathbf{u}}^{(t)}\right]\left(\int \phi_m^{(t)}(x')\mathbf{K}_{\mathbf{uu}}^{(t)-1}\mathbf{K}_{\mathbf{u}f_{x'}}^{(t)}dx'\right)} \\
&= \cancel{\left[\mathbf{K}_{\mathbf{uu}}^{(t)}\right]_{lm}} \cancel{- \left[\mathbf{K}_{\mathbf{uu}}^{(t)}\right]_{l,:}\mathbf{K}_{\mathbf{uu}}^{(t)-1}\mathbf{K}_{\mathbf{uu}}^{(t)}\mathbf{K}_{\mathbf{uu}}^{(t)-1}\left[\mathbf{K}_{\mathbf{uu}}^{(t)}\right]_{:,m}} + \left[\mathbf{K}_{\mathbf{uu}}^{(t)}\right]_{l,:}\mathbf{K}_{\mathbf{uu}}^{(t)-1}\mathbf{S}_{\mathbf{u}}^{(t)}\mathbf{K}_{\mathbf{uu}}^{(t)-1}\left[\mathbf{K}_{\mathbf{uu}}^{(t)}\right]_{:,m} \\
&= \left[\mathbf{S}_{\mathbf{u}}^{(t)}\right]_{lm},
\end{aligned}
\tag{C.9}
$$

这恰好是 $q(\boldsymbol{u}^{(t)})$ 中 $u_l^{(t)}$ 与 $u_m^{(t)}$ 之间的变分协方差。

因此，若 $f \sim q_t(f)$，则 $\int f(x)\phi_m^{(t)}(x)\mathrm{d}x \sim q(u_m^{(t)})$，这意味着我们可以用有限基近似后验 OHSVGP： $f = \sum_{m=1}^{M} u_m^{(t)} g_m^{(t)}(x)$， $u_m^{(t)} \sim q(u_m^{(t)})$。

C.3 SVGPVAE 模型详情

对于 SVGPVAE，我们采用Jazbec 等人 (2021年) 的方法和 Zhu 等人 (2023年中的符号，得到以下编码器-解码器模型：

$$p(\mathbf{y}_{1:T}) = p(\mathbf{f}_{1:T}) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{f}_t), \tag{C.10}$$

似然为 $p(\mathbf{y}_t|\mathbf{f}_t) = \mathcal{N}(\mathbf{y}_t|\varphi(\mathbf{f}_t)\ \sigma^2 I)$，解码器网络为 φ： $\mathbb{R}^L \to \mathbb{R}^{d_y}$。编码器 ϕ： $\mathbb{R}^{d_y} \to \mathbb{R}^{2L}$ yields $(\tilde{\mathbf{y}}_t^{1:L}, \tilde{\mathbf{v}}_t^{1:L}) = \phi(\mathbf{y}_t)$。$\mathbf{f}_t$ 服从一个 L 维多输出高斯过程，其近似后验为：

$$q(\mathbf{f}_{1:T}) = \prod_{l=1}^{L} p(\mathbf{f}_{1:T}^l|\mathbf{u}_m^l) q(\mathbf{u}^l), \quad q(\mathbf{u}^l) = \mathcal{N}(\mathbf{u}^l|\mathbf{m}^l, \mathbf{A}^l), \tag{C.11}$$

$$\mathbf{S}^l = \mathbf{K}_{\mathbf{uu}}^l + \mathbf{K}_{\mathbf{uf}}^l \mathrm{diag}(\tilde{\boldsymbol{v}}_{1:T}^l)^{-1} \mathbf{K}_{\mathbf{fu}}^l, \quad \mathbf{m}^l = \mathbf{K}_{\mathbf{uu}}^l (\mathbf{S}^l)^{-1} \mathbf{K}_{\mathbf{uf}}^l \mathrm{diag}(\tilde{\boldsymbol{v}}_{1:T}^l)^{-1} \tilde{\mathbf{y}}_{1:T}^l, \tag{C.12}$$

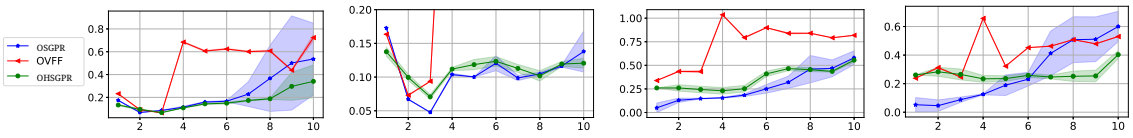
$$\mathbf{A}^l = \mathbf{K}_{\mathbf{uu}}^l (\mathbf{S}^l)^{-1} \mathbf{K}_{\mathbf{uu}}^l, \tag{C.13}$$

其中 $p(\mathbf{f}_{1:T}^l|\mathbf{u}_m^l)$ 是先验条件分布。

遵循 Jazbec 等人 (2021年) 的方法，目标函数定义如下：

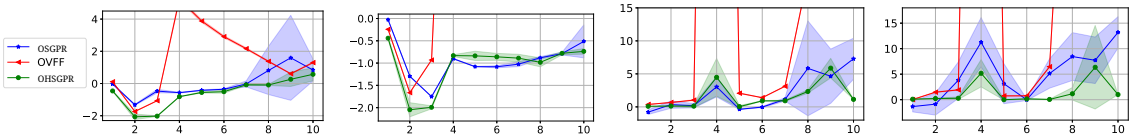
$$\mathcal{L}_{\mathrm{SVGPVAE}}(\theta) = \sum_{t=1}^{T} \left[\mathbb{E}_{q(\mathbf{f}_t)} \log p(\mathbf{y}_t|\mathbf{f}_t) - \log \mathcal{N}(\mathbf{f}_t|\tilde{\mathbf{y}}_t, \tilde{\mathbf{v}}_t) \right] + \sum_{l=1}^{L} \mathcal{L}_H^l, \tag{C.14}$$

其中 $\mathcal{L}_H^l$ 是 Jazbec 等人（2021年）在公式 7 中描述的“Hensman”证据下界。由于变分参数 $\mathbf{m}^l$ 和 $\mathbf{S}^l$，以及似然函数都由神经网络进行摊销，因此我们进一步在上述损失函数中加入针对编码器和解码器网络的 EWC（弹性权重巩固；（Kirkpatrick 等人，2017年））正则化，以实现持续学习。



(a) 太阳能，M=50　(b) 太阳能，M=150　(c) 音频，M=100　(d) 音频，M=200

图 C.1：太阳辐照度和音频信号预测数据集上，已学习任务的测试集均方根误差随已学习任务数量的变化（持续更新核超参数）。



(a) 太阳能，M=50　(b) 太阳能，M=150　(c) 音频，M=100　（d）音频，M=200

图 C.2：太阳辐照度和音频信号预测数据集上，已学习任务的测试集负对数预测密度随已学习任务数量的变化（持续更新核超参数）。

C.4 补充结果

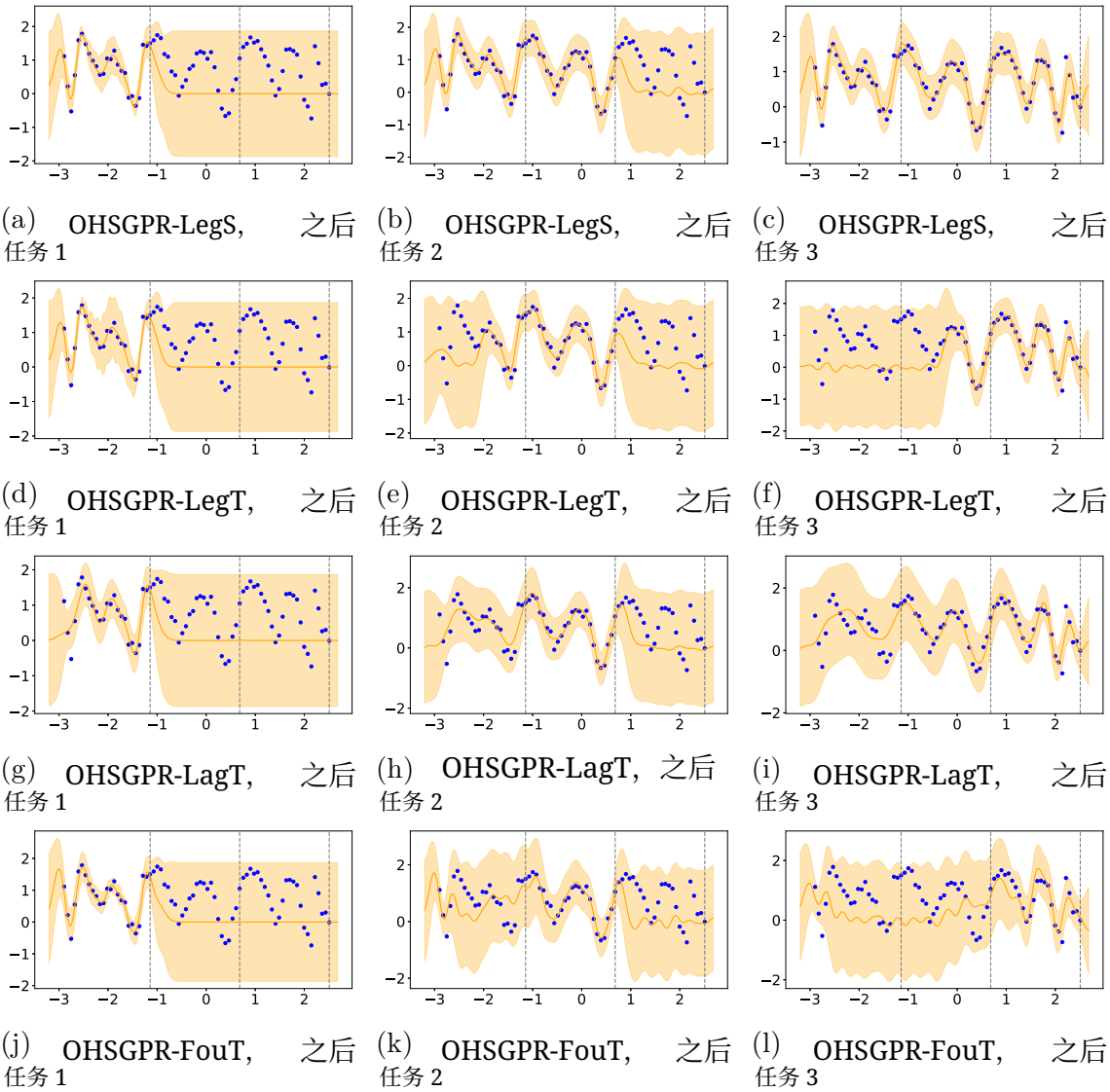
C.4.1 可训练核超参数下的时间序列回归结果

图 C.1 和 C.2 展示了基于可训练核超参数（即在所有任务中持续在线优化核超参数）的时间序列回归实验的均方根误差和负对数预测密度结果。请注意，OVC 仅适用于固定核（Maddox 等人，2021年），因此我们不在此处考虑它。与正文中基于固定核的结果相比，此处所有方法的表现都较不稳定。先前的研究要么找到性能良好的固定核（Maddox 等人，2021年），要么采用广义变分推断，通过正因子缩放在线证据下界目标中的 KL 项，并需谨慎调节该因子，以缓解核超参数在线优化的不稳定性（Stanton 等人，2021年；Kapoor 等人，2021年）。

C.4.2 基函数–测度变体的比较

图 C.3展示了 OHSGPR 应用于玩具时间序列回归数据集的结果。数据按时间顺序分为三个相等的片段，并在在线学习设置中分别作为任务 1–3。该图比较了用于 OHSGPR 的若干 HiPPO 算子变体（Gu 等人，2020年、2023年）。子图 (a–c) 对应于我们所有主要实验中使用的 HiPPO-LegS。子图 (d–f) 使用 HiPPO-LegT，(g–i) 使用基于拉盖尔

多项式基的 HiPPO-LagT，而 (j–l) 使用基于傅里叶基函数的 HiPPO-FouT。这些 HiPPO 变体的详细公式见第 2.1.2节和附录 A.1。OHSGPR-LegS 成功记住了所有过去的任务，而 OHSGPR-LegT、OHSGPR-LagT 和 OHSGPR-FouT 都表现出一定程度的灾难性遗忘。这是因为它们没有像 HiPPO-LegS 那样对过去采用均匀测度，而是基于对近期历史赋予更大权重的测度。LegT 和 FouT 使用固定长度滑动窗口测度，而 LagT 使用指数衰减测度，对近期历史赋予更高的重要性。



(a) OHSGPR-LegS, 之后任务 1　(b) OHSGPR-LegS, 之后任务 2　(c) OHSGPR-LegS, 之后任务 3

(d) OHSGPR-LegT, 之后任务 1　(e) OHSGPR-LegT, 之后任务 2　(f) OHSGPR-LegT, 之后任务 3

(g) OHSGPR-LagT, 之后任务 1　(h) OHSGPR-LagT, 之后任务 2　(i) OHSGPR-LagT, 之后任务 3

(j) OHSGPR-FouT, 之后任务 1　(k) OHSGPR-FouT, 之后任务 2　(l) OHSGPR-FouT, 之后任务 3

图 C.3：基于不同 HiPPO 变体的 OHSGPR 在一个玩具在线回归数据集上的比较。

附录 D

第 5章的补充材料

D.1 定理 5.4.1的证明

证明。 为推导 $\hat{\boldsymbol{z}}_T^{(g)*}(\boldsymbol{z}_{s_1}, \cdots \boldsymbol{z}_{s_k}) = \arg\min_{\hat{\boldsymbol{z}}_T^{(g)}} \mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{s_1}, \cdots \boldsymbol{z}_{s_k})}[||\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(g)}||_2^2]$, 我们首先计算梯度,

$$\begin{aligned}\nabla_{\hat{\boldsymbol{z}}_T^{(g)}} \mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[||\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(g)}||_2^2] &= \mathbb{E}_{p(\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}\{\nabla_{\hat{\boldsymbol{z}}_T^{(g)}} \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[||\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(g)}||_2^2]\} \\ &= 2\mathbb{E}_{p(\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}\{\mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(g)}]\}.\end{aligned} \tag{D.1}$$

令上述梯度为 0, 可得

$$\mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(g)}] = 0 \implies \hat{\boldsymbol{z}}_T^{(g)*}(\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k}) = \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[\boldsymbol{z}_T]. \tag{D.2}$$

将 $\hat{\boldsymbol{z}}_T^{(g)*}(\boldsymbol{z}_{s_1}, \cdots \boldsymbol{z}_{s_k})$ 代入$\mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{s_1}, \cdots \boldsymbol{z}_{s_k})}[||\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(g)}||_2^2]$, 即可得到最小重建误差。

$$\begin{aligned}\min_{\hat{\boldsymbol{z}}_T^{(g)}} \mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[||\boldsymbol{z}_T - \hat{\boldsymbol{z}}_T^{(g)}||_2^2] &= \mathbb{E}_{p(\boldsymbol{z}_T, \boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[||\boldsymbol{z}_T - \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[\boldsymbol{z}_T]||_2^2] \\ &= \mathbb{E}_{p(\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1}, \cdots, \boldsymbol{z}_{s_k})}(\boldsymbol{z}_T)].\end{aligned} \tag{D.3}$$

同理，

$$\begin{aligned}\min_{\hat{\boldsymbol{z}}_T^{(h)}} \mathbb{E}_{p(\boldsymbol{z}_T,\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}[||\boldsymbol{z}_T-\hat{\boldsymbol{z}}_T^{(h)}||_2^2] &= \mathbb{E}_{p(\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}(\boldsymbol{z}_T)],\\ \hat{\boldsymbol{z}}_T^{(h)*}(\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l}) &= \mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}[\boldsymbol{z}_T],\end{aligned} \tag{D.4}$$

现在，我们通过观察 $T>s_1>\cdots>s_k>\cdots>s_l$，说明随着作为输入的先前帧增多，重建误差绝不会增加。

$$\begin{aligned}&\min_{\hat{\boldsymbol{z}}_T^{(h)}}\mathbb{E}_{p(\boldsymbol{z}_T,\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}[||\boldsymbol{z}_T-\hat{\boldsymbol{z}}_T^{(h)}||_2^2] = \mathbb{E}_{p(\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}(\boldsymbol{z}_T)]\\ &\quad\le \mathbb{E}_{p(\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}(\boldsymbol{z}_T)] = \min_{\hat{\boldsymbol{z}}_T^{(g)}} \mathbb{E}_{p(\boldsymbol{z}_T,\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}[||\boldsymbol{z}_T-\hat{\boldsymbol{z}}_T^{(g)}||_2^2].\end{aligned} \tag{D.5}$$

事实上，

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}(\boldsymbol{z}_T)] - \mathbb{E}_{p(\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}(\boldsymbol{z}_T)]\\ &= \mathbb{E}_{p(\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}\{\mathbb{E}_{p(\boldsymbol{z}_{s_{k+1}},\cdots,\boldsymbol{z}_{s_l}|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}[\mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}(\boldsymbol{z}_T)] - \mathrm{Var}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}(\boldsymbol{z}_T)\}\\ &= -\mathbb{E}_{p(\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}\{\mathrm{Var}_{p(\boldsymbol{z}_{s_{k+1}}\cdots,\boldsymbol{z}_{s_l}|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}(\mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}[\boldsymbol{z}_T])\} \qquad \text{(Law of total variance)}\\ &\le 0.\end{aligned} \tag{D.6}$$

请注意，如果 $\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots\boldsymbol{z}_{s_k}\overset{d}{=}\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots\boldsymbol{z}_{s_l}$，那么上述差值将变为 0：

$$\begin{aligned}&\mathrm{Var}_{p(\boldsymbol{z}_{s_{k+1}}\cdots,\boldsymbol{z}_{s_l}|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}(\mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_l})}[\boldsymbol{z}_T])\\ &= \mathrm{Var}_{p(\boldsymbol{z}_{s_{k+1}}\cdots,\boldsymbol{z}_{s_l}|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}(\mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots,\boldsymbol{z}_{s_k})}[\boldsymbol{z}_T])\\ &= 0,\end{aligned} \tag{D.7}$$

因为 $\mathbb{E}_{p(\boldsymbol{z}_T|\boldsymbol{z}_{s_1},\cdots\boldsymbol{z}_{s_k})}[\boldsymbol{z}_T]$ 仅是 $\boldsymbol{z}_{s_1},\cdots\boldsymbol{z}_{s_k}$的函数。

因此，要使不等式严格成立，就必须避免 $\boldsymbol{z}_T \perp\!\!\!\perp \boldsymbol{z}_{s_{k+1}},\cdots\boldsymbol{z}_{s_l}\mid\boldsymbol{z}_{s_1},\cdots\boldsymbol{z}_{s_k}$，其中一阶马尔可夫链（$\boldsymbol{z}_T\to\cdots\boldsymbol{z}_{s_k}\to\boldsymbol{z}_{s_l}$）是其特例。

□

D.2 超参数

在实验中，我们首先选择可在我们的图形处理器（NVIDIA RTX A6000）上训练的最大帧数；CViViT 和视频扩散分别为 18 帧和 8 帧。随后，我们考虑纳入

帧数约为最大值一半时的结果（CViViT 和视频扩散分别为 8 帧和 4 帧）。我们的实验超参数如下表所示。

表 D.1：C-ViViT 架构和优化器所使用的超参数。

	1帧	8帧	18帧
空间层数	8	4	4
时间层数	-	4	4
嵌入维度	512	512	512
隐藏维度	512	512	512
头数	8	8	8
学习率	1e-4	1e-4	1e-4
学习率调度器	余弦衰减	余弦衰减	余弦衰减
训练步数	100k	100k	100k
批次大小	64	64	64

表 D.2：VideoGPT架构和优化器所用的超参数。

	1帧	8帧	18帧
层数	8	8	8
嵌入维度	144	144	144
隐藏维度	144	144	144
头数	4	4	4
学习率	1e-4	1e-4	1e-4
学习率调度器	余弦衰减	余弦衰减	余弦衰减
训练步数	100k	100k	100k
批次大小	64	64	64

表 D.3：Phenaki 架构和优化器所使用的超参数。

	1帧	8帧	18帧
层数	6	6	6
嵌入维度	512	512	512
隐藏维度	512	512	512
头数	8	8	8
学习率	1e-4	1e-4	1e-4
学习率调度器	余弦衰减	余弦衰减	余弦衰减
训练步数	200k	200k	200k
批次大小	64	64	64

表 D.4：视频扩散的 UNet 架构和优化器所用的超参数。

	1帧	4帧	8帧
下采样/上采样层的数量	4	4	4
残差块的数量	2	2	2
基础通道大小	128	128	128
每生成一帧的扩散步数	1000	1000	1000
学习率	1e-4	1e-4	1e-4
训练步数	500k	500k	500k
批次大小	32	32	32