这段文字讨论了基于\*\*正态流（Normalizing Flows, NFs）\*\*的无损数据压缩方法，以及不同压缩算法的优缺点。

### 1. ****正态流（NFs）和数据分布推断****

* **正态流（NFs）利用变化变量公式**（change-of-variable formula）通过设计**双射映射（bijective maps）**，将数据（x）和潜在表示（z）之间的关系明确建模，从而进行精确的数据分布推断。
* **无损压缩**：正态流通过这种映射与**熵编码算法**结合，实现了数据的无损压缩。具体来说，NFs在进行压缩时可以通过模型来推断数据的分布，并利用该分布进行高效编码和解码。

### 2. ****与IDF的对比****

* **IDF（Integer Discrete Flows）是另一种基于正态流的模型，它与上述方法不同的是，IDF模型将数据x和潜在表示z都建模为离散整数**。
* \*\*Ho et al. (2019b)**提出了一种改进方法，离散化连续变量，并提出了**局部回退位（LBB）\*\*方案，允许在使用正态流的同时进行无损压缩，尤其是当x和z是连续变量时。

### 3. ****连续NFs面临的问题****

* **连续NFs（如Ho et al. 2019b的模型）的问题是数值不可逆**，这使得数据压缩过程中的推断变得更复杂且较慢。Zhang等人（2021c; b）提出了**数值可逆的连续流变换**，解决了这一问题。

### 4. ****与其他模型的比较****

* **分层变分自编码器（Hierarchical VAEs）**：这些模型的理论编码长度与正态流模型类似（例如Ho等人，2019a）。它们在压缩率上与NFs相当。
* **DPM（Denoising Probability Models）**：DPM在数据压缩中的表现非常出色，尤其是在**CIFAR10、ImageNet32、ImageNet64**等数据集上，能够达到较低的编码长度（例如：CIFAR10为2.49 bits/dim）。
* 然而，尽管这些模型的**理论编码长度**表现优秀，它们的推理过程和编码过程通常更加**复杂且缓慢**，例如，Kingma等人（2021）提出的方法需要大量的计算和时间步骤来进行推理。

### 5. ****压缩速度与实用性****

* **IDF的优势**在于其**效率高**，即使压缩率略有损失，它仍然是目前最高效的神经网络压缩方法。
* 但**IDF的带宽较低**，约为1MB/s，无法满足某些实际应用中的需求。
* **Bits-back编码**方案是连续流、变分自编码器（VAE）和DPM等方法的关键，但它们在压缩少量数据时会产生**额外的辅助位**，增加了额外的开销。

### 6. ****其他模型的性能****

* **ARM（Autoregressive Models）的理论编码长度与DPM相似，但其解码速度极慢**，因为它采用的是**串行采样**过程。

### 总结：

* **IDF**被认为是最高效的神经网络压缩器，但它在压缩率上稍微逊色，且带宽限制较低。
* **NFs**（尤其是改进的连续NFs）在理论上能够提供更好的压缩率，但它们的推理和编码过程复杂且慢，适用于数据量较大的场景。
* **DPM**和**VAEs**等模型虽然在压缩率上有优势，但也面临着计算复杂性和推理速度较慢的挑战。

总之，虽然正态流和其他深度生成模型在压缩效率和理论编码长度上有所提升，但目前的压缩方法（尤其是对于大规模数据）仍面临着推理速度和复杂度的瓶颈，IDF则是一个较为实用且高效的压缩方案。