arXiv:2603.00949v1 [cs.CV] 1 Mar 2026

StegoNGP：基于Instant-NGP的3D密码学隐写

Wenxiang Jiang Yujun Lan 山东工商学院 中国海洋大学jwx@sdtbu.edu.cn, lanyujun4733@stu.ouc.edu.cnShuo Zhao Yuanshan Liu 中国海洋大学 山东工商学院 zhaoshuo@ouc.edu.cn yshliu@sdtbu.edu.cnMingzhu Zhou * Jinxin Wang *阿德莱德大学 山东工商学院mingzhu.zhou@student.adelaide.edu.au wangjinxin@sdtbu.edu.cn

摘要

近年来，即时神经图形基元（*Instant-NGP*）在快速*3D*场景重建方面取得了显著成功，但如何安全地嵌入高容量隐藏数据（例如完整的三维场景）仍然是一项挑战。现有方法依赖外部解码器，需要架构修改，且容量有限，因而容易被检测。我们提出一种新颖的无参数3D密码学**隐**写，**基于**Instant-**NGP**（StegoNGP），它将*Instant-NGP*的哈希编码函数用作密钥控制的场景切换器。通过将默认密钥与载体场景相关联，并将秘密密钥与隐藏场景相关联，我们的方法训练单一模型，使两种表示交织在相同的网络权重中。所得模型在架构和参数量上与标准*Instant-NGP*无法区分。我们还引入了一种增强的多密钥方案，在各哈希层级上分配多个独立密钥，从而大幅扩展密钥空间，并对部分密钥泄露攻击具有很强的鲁棒性。实验结果表明，*StegoNGP*能够以很强的不可感知性和安全性隐藏完整的高质量三维场景，为神经场中高容量、不可检测的信息隐藏提供了新范式。代码参见 *[*链接将在录用后公开*]*。

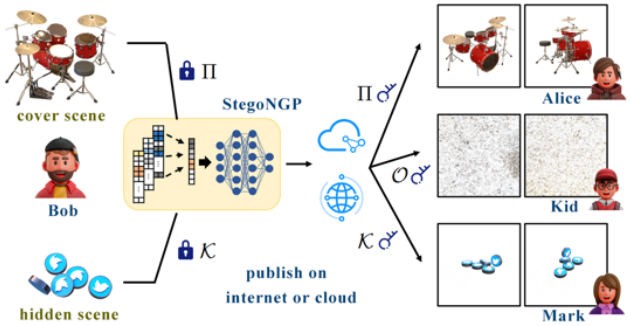

图 1. 所提出的 StegoNGP 的应用场景：三维模型创作者 Bob 创建一个三维场景（载体场景），并使用隐写密钥 $\mathcal{K}$将隐藏信息（例如版权信息）嵌入其中。随后，训练后的 StegoNGP 会公开发布在互联网上或云盘中，其模型与标准的 Instant-NGP 模型无法区分。普通用户 Alice 可以使用并渲染原始载体场景。持有密钥 $\mathcal{K}$的预期接收者 Mark 可以提取隐藏场景，从而实现版权保护或秘密通信。试图使用无关密钥 $\mathcal{O}$访问隐藏场景的 Kid 只能获得毫无意义的空白图像或噪声图像。

1. 引言

神经辐射场（NeRF） [14]通过使用多层感知机（MLP）隐式学习 3D 表示，显著推动了 3D 重建领域的发展。在瞬时神经图形基元（Instant-NGP） [15] 和 3D 高斯泼溅（3DGS） [10] 通过采用混合显式-隐式表示，进一步提升了实时渲染能力、保真度和易用性。随着这些三维模型的数量不断增加，它们作为数字资产的价值也迅速攀升。因此，人们对安全且隐蔽地隐藏高容量信息的需求日益增长，应用场景包括密码学版权保护、可追踪元数据嵌入和秘密通信。这推动了三维隐写等领域的研究

[7, 12, 19] 与水印 [4, 11, 23]。

传统上，大多数 NeRF 水印技术都专注于将二进制消息等低容量数据 [3, 18], 嵌入模型，以保护版权。相比之下，StegaN-eRF [11]和 NeRF 隐写术 [4] 等 NeRF 隐写方法可以将二维图像嵌入三维模型。然而，这些方法容量有限，需要外部解码器，也无法支持安全通信。IPA-NeRF [9] 和 GaussTrap [6] 利用秘密视点作为后门触发器，将图像嵌入 NeRF 和 3DGS 模型的特定视角中，但这种方法同样容量有限，而且实际上会损害三维模型的完整性。

在诸如 Instant-NGP [15], 等紧凑的三维神经辐射场表示中隐藏高容量加密信息，同时保持通信隐私与安全性，仍然是一项重大挑战，尤其是在不改变三维模型架构、也不增加额外参数或解码器的情况下。

本文提出一种三维密码隐写术，基于 *Instant-NGP*（*StegoNGP*），以应对这一挑战。我们将 Instant-NGP 的哈希映射函数转换为密钥控制的场景切换器。Instant-NGP 中的原始哈希函数依赖一个由大素数组成的默认序列，将三维坐标映射到特征表。我们将该素数序列视为默认密钥 Π。随后，我们引入一个新的秘密素数序列$\mathcal{K}$，作为隐写密钥。我们训练单个 Instant-NGP 模型，以同时优化两个场景：载体场景 $\mathcal{S}$ 使用密钥 Π，以及隐藏场景 $\mathcal{B}$ 使用密钥$\mathcal{K}$。这使我们能够以零参数开销将一个完整的三维场景隐藏在另一场景的模型中，且仅可通过密码密钥访问。应用场景如图1。

值得注意的是，GS-Hider [23]通过向每个各向异性高斯添加额外的球谐参数，并使用图像解码器，实现了基于 3DGS 模型 [10] 的三维场景隐写术。相比之下，我们的研究并未向 Instant-NGP 添加额外参数或结构。因此，一旦公开发布，由我们的方法生成的模型将与标准 Instant-NGP 无法区分。总而言之，我们的工作在以下方面推进了该领域的发展：

• 我们提出了 StegoNGP，一种新颖的无参数三维密码隐写术框架。通过利用基于哈希的神经场固有的结构冗余，我们的方法无需辅助解码器或架构修改，即可实现高容量数据隐藏。

•我们设计了一种独特的密钥控制的场景切换机制，利用秘密素数序列隐藏一个完整的高质量三维场景。该方法具有很强的不可感知性，因为载体场景的质量不受影响，同时密钥的保密性确保了安全性。

•我们提出了更为稳健的方案 Multi-Key StegoNGP，为不同的哈希层级分配多个相互独立的密钥。这显著扩展了密钥空间并增强了安全性，使其能够高度抵抗部分密钥泄露攻击，因为攻击者必须掌握完整的密钥集才能恢复隐藏场景。

2. 相关工作

2.1. 神经三维场景表示

神经辐射场（NeRF） [14]利用多层感知机网络，从一组二维图像中学习连续的隐式 3D表示，并能够从未见视点合成逼真的新视图。在 NeRF 的基础上，后续研究迅速推动了这一领域的发展，并在多个方向取得了显著进展。即时神经图形基元（Instant-NGP） [15] 提出了一种多分辨率 3D 哈希特征嵌入方法，该方法将三维空间坐标映射为存储在多级特征哈希表中的可训练特征向量。随后，小型 MLP 对拼接后的多级特征进行解码，得到颜色和密度，从而大幅提升训练和渲染速度。同时，作为显式表示三维场景的另一种改进方法，3D 高斯泼溅（3DGS）[10], 利用数百万个三维高斯表示场景，同样实现了高质量渲染和实时渲染速度。其他相关工作还在降低模型复杂度 [13, 17], 、提升渲染质量 [1, 2], 、扩展到更大场景 [5, 20],，甚至将 NeRF 拓展到动态场景 [16, 22]等方面取得了进展。

2.2. NeRF水印与隐写术

随着三维重建技术不断进步，NeRF及其变体生成的模型作为数字资产正不断积累价值。因此，对版权保护、元数据嵌入和信息隐藏的需求日益增长。这推动了 NeRF水印 [7, 12, 18, 19] 和隐写术 [4, 11, 23]这两个相关领域的研究。

NeRF水印。 NeRF水印主要关注版权保护和所有权验证。这些方法通常专用于嵌入容量较低但稳健的信息，例如二进制消息[3, 7, 8, 18]或小型标识 [19]。这些水印经过设计，即使模型经过压缩或微调等变换后，仍然可以被识别。此类方法的容量通常较小，例如 16 位 [7], 48 位 [3, 18]或 64 位 [8]二进制信息，并且需要专用解码器，这显著限制了秘密信息传输容量的上限。

NeRF 隐写术。 相比之下，NeRF 隐写术旨在实现高容量和不可感知性， $i.e.$，在不引起怀疑的情况下隐藏大量数据。Ste-gaNeRF [11] 或 NeRF 隐写术 [4] 探索了将二维图像嵌入 NeRF 模型。然而，这些方法不仅容量受限于二维数据，而且通常需要额外的外部解码器网络来提取隐藏信息，这使得隐写术的存在在实践中很容易被察觉。

最近，GS-Hider [23]取得了重大进展。它基于 3DGS 模型 [10]，通过为每个高斯添加额外的球谐（SH）参数和一个辅助解码器，成功隐藏了另一个完整的三维场景。尽管 GS-Hider 实现了高容量隐藏，但其方法引入了显著的参数开销，并改变了模型的基础架构。这种架构修改和参数量的增加使模型在结构上可被检测（$i.e.$，可通过参数量或处理流程结构检测），违背了隐写术的核心目标。

2.3. 其他 NeRF 信息隐藏方法

除了上述讨论的水印和隐写术方法外，IPA-NeRF [9] 和 GaussTrap [6] 利用秘密视点作为后门触发器，在模型中隐藏信息。具体而言，当模型从普通视点渲染时，会显示原始载体场景；然而，当从预设的秘密视点渲染时，则会显现隐藏的后门图像。

尽管这种方法在概念上十分优雅，且避免了额外解码器带来的开销，但仍存在局限性。首先，其隐藏容量通常仅限于一张二维图像，因此无法隐藏整个三维场景。其次，这种方法实际上可能会局部损害三维模型的完整性。这与我们的 StegoNGP 有本质区别：StegoNGP 使用的密码学哈希密钥与模型渲染完全解耦。

3. 方法

3.1. 预备知识

本节首先回顾 NeRF 渲染和 Instant-NGP 多分辨率哈希特征嵌入的基础数学定义。这些定义为后续的问题定义以及所提出的 StegoNGP 方法奠定基础。

神经辐射场（NeRF）。对于 NeRF 模型 F：$(\mathbf{x},\ \mathbf{d}) \rightarrow (c,\ \tau)$，输入是一个五元组，即采样点 $\mathbf{x} \in \mathbb{R}^3$ 的坐标以及采样射线 $\mathbf{d} \in \mathbb{R}^3$的方向，输出是一个 RGB 颜色 $c \in [0,1]^3$ 和一个体密度 $\tau \in \mathbb{R}^+$。目标图像中的每个像素表示空间中的一条射线 $\mathbf{x} = \mathbf{r}(t)$：$= \mathbf{o} + t\mathbf{d}$，它从相机中心 $\mathbf{o}$ 出发，朝射线方向 $\mathbf{d}$延伸，其中 t 表示射线深度。沿着射线方向，对离散采样点 n 的输出进行积分，得到像素的预测颜色值 C，如下所示：

$$\hat{C}(\mathbf{r}, F) := \sum_{i=1}^{n} T(t_i) \cdot \alpha(\tau(t_i) \cdot \delta_i) \cdot c(t_i), \tag{1}$$

$$T(t_i) := \exp(-\sum_{j=1}^{i-1} \tau(t_j) \cdot \delta_j), \tag{2}$$

其中，$\alpha(x) := 1 - \exp(-x)$，$\delta_i := t_{i+1} - t_i$ 表示两个采样点之间的距离，$c(t_i)$ 和 $\tau(t_i)$ 分别表示$\mathbf{r}(t_i)$处的颜色和密度。

即时神经图形基元（Instant-NGP）。Instant-NGP 通过使用多分辨率三维哈希特征嵌入和浅层 MLP 来表示三维场景，其训练和渲染速度比原始 NeRF 快数倍。具体而言，Instant-NGP 将空间划分为多个分辨率级别的网格。每一级的分辨率 N_l 会在最粗级别 N_{min} 与最精细级别 N_{max} 之间按几何级数变化，如下所示：

$$N_l := [N_{min} \cdot z^l], \tag{3}$$

$$z := \exp\left(\frac{\ln N_{max} - \ln N_{min}}{L-1}\right). \tag{4}$$

对于给定的采样位置$\mathbf{x}$，Instant-NGP 模型$\mathcal{M}$：$(\mathcal{T}(h(\mathbf{x})),\ \mathbf{d}) \rightarrow (c,\ \tau)$ 使用一个四层 MLP，将编码映射为 RGB 颜色 c和体密度 τ。该编码由多个分辨率级别的特征向量拼接而成，这些向量通过哈希表查找 $\mathcal{T}(\cdot)$ 获得。哈希表索引由哈希函数 $h(\mathbf{x})$ 根据采样位置 $\mathbf{x}$ 计算得出，如下所示：

$$h(\mathbf{x}) := \left(\bigoplus_{i=1}^{d} x_i \pi_i\right) \mod T, \tag{5}$$

其中，$\oplus$ 表示按位异或运算，π_i 是互不相同的大素数，d 表示输入坐标$\mathbf{x}$的维度，而 T 是每级的最大条目数（$i.e.$，即哈希表大小）。

3.2. 问题表述

考虑一种具备双重功能的密码隐写模型 $\mathcal{H}$。它会加密并嵌入隐藏场景 B，输入加密密钥$\mathcal{K}$即可对其解码；而通过其默认流程 $\mathcal{M}$ 进行渲染时，若不提供

密钥，则会呈现原始封面场景 S。因此，密码隐写的目标如下：

$$\min_{\mathcal{H}} \forall_{v \in V} (\|\mathcal{I}(\mathcal{H}(\mathcal{K}), v) - b_v\|_2^2 + \|\mathcal{I}(\mathcal{M}, v) - s_v\|_2^2), \tag{6}$$

其中，V是观测视点的集合，$\mathcal{I}$表示合成函数，它整合沿射线ˆ的采样点所获得的颜色和体密度，以合成视点 $v \in V$的预测图像 I_v；而 $b_v \in B$ 和 $s_v \in S$ 分别是从视点 v观察隐藏场景 B 和载体场景 S 得到的真实值图像。设 $\mathcal{M}$为标准 Instant-NGP 模型，它是在不存在解密密钥时，使用默认大素数序列 Π（$i.e.$、$\mathcal{M} \equiv \mathcal{H}(\Pi)$）时，$\mathcal{H}$ 的退化形式。

3.3. 密码隐写

对于 $d = 3$，原始 Instant-NGP 模型使用默认素数序列 $\Pi \equiv \{1, 2654435761, 805459861\}$。我们将 $h(\mathbf{x})$ 推广为密钥哈希索引函数 $\psi(\mathbf{x}\ \mathcal{K})$，以秘密密钥 $\mathcal{K} \equiv \{k_1, k_2, k_3\}$替换 Π，定义如下：

$$\psi(\mathbf{x}, \mathcal{K}) := \left(\bigoplus_{i=1}^{d} x_i k_i \right) \mod T. \tag{7}$$

此外，通过在默认序列与给定的秘密序列（例如 $\mathcal{K} = \{54857899, 1455645677, 5487678709\}$）之间切换加密密钥，我们可以同时将两个场景嵌入模型中：载体场景 $\mathcal{S}$ 和隐藏场景 $\mathcal{B}$。StegoNGP $\mathcal{H}$经过优化以解决方程6中的问题，其定义如下：

$$\mathcal{H}(\mathcal{K}, \mathcal{S}, \mathcal{B}) := \begin{cases} \mathcal{M}(\mathcal{T}(\psi(\mathbf{x}, \mathcal{K})), \mathbf{d}), & \text{for hidden scene } \mathcal{B} \\ \mathcal{M}(\mathcal{T}(h(\mathbf{x})), \mathbf{d}), & \text{for cover scene } \mathcal{S}. \end{cases} \tag{8}$$

如算法1所示，每次训练迭代都会随机选择载体场景 S 或隐藏场景 B 进行拟合。载体场景使用默认的大素数序列 Π，而隐藏场景使用加密密钥 $\mathcal{K}$。计算均方误差损失 $\mathcal{L}_{mse}$和权重稀疏性损失 $\mathcal{L}_{spar}$，并用它们训练 StegoNGP $\mathcal{H}$。

图2 直观地展示了我们提出的 StegoNGP 框架，其中 MLP 网络利用共享哈希表中的特征来渲染场景 $\mathcal{T}$ 以及辅助输入 ξ（例如编码后的视角方向）。其核心

算法1 密码隐写

输入： S：载体场景，B：隐藏场景
输入： Π：默认的大素数列表
输入： $\mathcal{K}$：隐写加密密钥
输入： $\mathcal{M}$：初始化Instant-NGP模型
输入： β：损失权重，η：学习率
输入： K：最大迭代次数
输出： $\mathcal{H}$：同时包含封面和隐藏场景

$\mathcal{H} \leftarrow \mathcal{M}$
while $i < K$ **do**
  $\mathcal{S}^* \leftarrow random.choice([S, B])$
  if $\mathcal{S}^*$ is S **then**
    $\mathcal{K}^* \leftarrow \Pi$
  else
    $\mathcal{K}^* \leftarrow \mathcal{K}$
  end if
  $I_v, v \leftarrow random.choice(\mathcal{S}^*)$
  $\hat{I}_v \leftarrow \mathcal{I}(\mathcal{H}(\mathcal{T}(\psi(\mathbf{x}, \mathcal{K}^*))), v)$
  $\mathcal{L} \leftarrow \mathcal{L}_{mse}(\hat{I}_v, I_v) + \beta \cdot \mathcal{L}_{spar}(\mathcal{H})$
  $\mathcal{H} \leftarrow \mathcal{H} + \eta \cdot \nabla_{\mathcal{H}} \mathcal{L}$
end while

是我们的框架采用的密钥控制哈希函数，它会查询表 $\mathcal{T}$。通过标准哈希函数 $h(\mathbf{x})$ 使用默认的大素数序列 Π 时，框架会重建或渲染载体场景 $\mathcal{S}$。相反，当向哈希函数 $\psi(\mathbf{x}\ \mathcal{K})$ 提供秘密密钥 $\mathcal{K}$ 时，该函数会计算出不同的索引，使完全相同的网络参数能够重建或渲染隐藏场景 $\mathcal{B}$。

3.4. 多密钥级别分配方案

虽然第 3.3 节所述的方法已提供稳健的密码学安全性，但对于需要更高保护级别的应用，其安全性保证仍可进一步增强。我们将这种增强配置称为 Multi-Key StegoNGP，它采用多密钥级别分配方案。在标准 Instant-NGP 中，有 $L = 16$ 个分辨率级别，默认哈希函数 $h(\mathbf{x})$在所有级别中均匀使用相同的大素数序列 Π。我们的基础 StegoNGP（见第 3.3 节）将其替换为单个秘密密钥 $\mathcal{K}$，该密钥也均匀应用于所有 L 个级别。我们可以通过定义一个包含 $\mathbf{K}$ 个 m 不同秘密密钥的密钥集来增强这一方案：

$$\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_m\}, \quad \text{where } 1 \leq m \leq L, \tag{9}$$

我们不再对所有级别使用同一个密钥，而是以循环方式将这些 m密钥分配给 L 级别。设 $l \in \{1, \ldots, L\}$为级别索引。分配给级别 l的特定密钥 $\mathcal{K}^{(l)}$

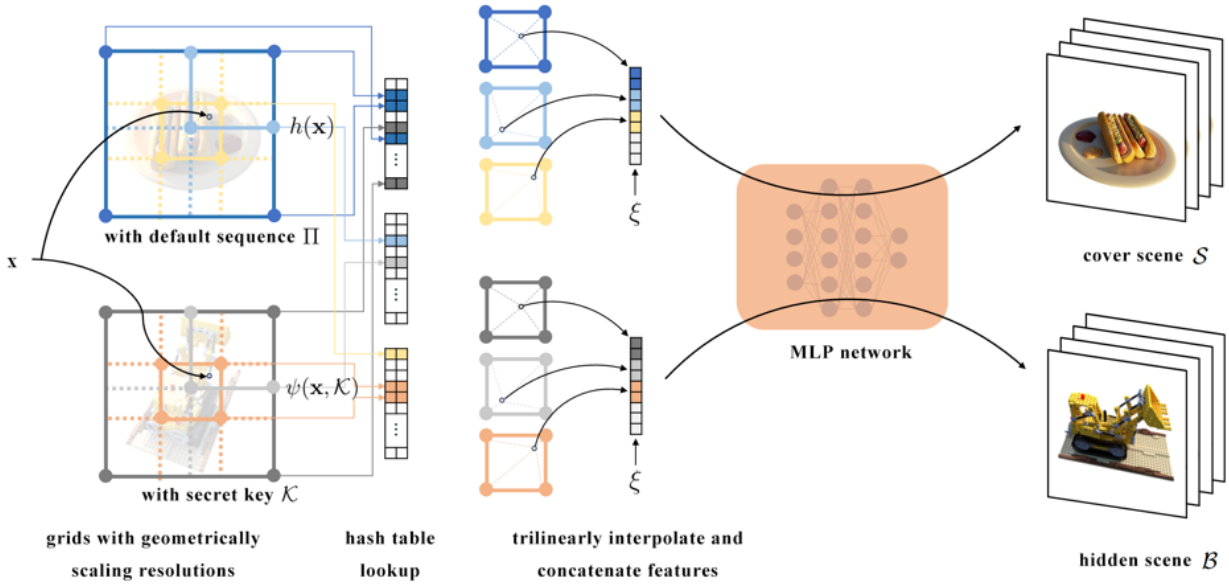

图2. 用于双场景表示的StegoNGP框架。

由其索引对 m取模确定：

$$\mathcal{K}^{(l)} = \mathcal{K}_j, \quad \text{where} \quad j = ((l-1) \mod m) + 1. \quad (10)$$

这种映射支持灵活且增强的安全性配置：

• **当 $\mathbf{m} = 1$**：密钥集为$\mathbf{K} = \{\mathcal{K}_1\}$，此时退化为我们的基础StegoNGP。
• **当 $\mathbf{m} = 2$**：密钥$\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2\}$ 分别应用于奇数层和偶数层。

• **当 $\mathbf{m} = \mathbf{L} = 16$**：这提供了最高安全性配置，其中16个层级各自采用唯一且独立的秘密密钥。

这种多密钥方法显著扩大了密码学密钥空间。攻击者不再能仅通过发现单个密钥 $\mathcal{K}$就攻破整个隐藏场景。他们必须发现全部 m 个密钥以及正确的分配顺序，从而使该模型对密码学分析和暴力破解攻击具有更强的稳健性。

4. 实验

在本节中，我们首先详细说明实验的基础设置。随后，我们通过与基线 Instant-NGP 模型进行比较，展示我们的方法在合成场景上的有效性。接下来，我们分析密钥敏感性与合理否认性。在此之后，我们将其与现有的无加密三维隐写方法进行比较。最后，我们探索其在真实世界场景上的泛化能力，并对多密钥方案进行全面的消融研究。

4.1. 实验设置

数据集。 所有实验均在 Blender 合成数据集和 Mip-Nerf-360 数据集上进行。Blender 合成数据集在原始 NeRF 论文中提出 [14], , 包含八个物体。每个物体由400 张从不同视点生成的图像组成，这些视点采样于上半球，分辨率为 800×800 像素，并被划分为 100 张图像用于训练，200 张用于测试，以及 100 张用于验证。此外，我们使用 Blender 以相同格式合成了两个额外的三维场景（Google 和 Twitter），作为我们的隐藏场景。Mip-Nerf-360 数据集 [2]包含九个真实世界采集的场景。为了更好地反映真实应用场景，其训练、测试和验证集之间的数据分布并不均匀。对于该数据集，我们对其图像进行预处理，并通过 COLMAP 生成相机位姿。

模型。 我们的基线模型是标准的 Instant-NGP[15], , 训练 50,000 次迭代。我们的 StegoNGP 采用相同的架构和总迭代次数（50,000 次）。每次迭代时，它会随机选择载体场景 $\mathcal{S}$ （使用默认密钥 Π）或隐藏场景 $\mathcal{B}$ （使用密钥 $\mathcal{K}$ 或多密钥$\mathbf{K}$）进行处理。此外，我们还将 StegoNGP 与 GS-Hider[23] 进行比较，后者采用默认配置，迭代次数上限同为 50,000 次。

4.2. 合成场景上的性能

我们在 Blender 合成数据集上评估了 StegoNGP框架的隐写效果。我们将方法与 Instant-NGP（基线）进行比较，以展示其在保持较高重建保真度的同时隐藏完整三维场景的能力。

在本实验中，我们采用 Blender 数据集中的八个标准物体作为载体场景（$\mathcal{S}$），并将两个自制合成场景（Google 和 Twitter）用作隐藏场景（$\mathcal{B}$）。我们通过将每个载体场景分别与两个隐藏场景之一配对，组成了八组实验。具体而言，Lego、Mic、Chair 和 Ficus 这四个场景与 Twitter 配对，另外四个场景（Hotdog、Drums、Ship 和 Materials）则与 Google 配对。我们分别在这 10 个场景上独立训练 Instant-NGP（基线）模型，以确定重建质量的上限。我们的 StegoNGP 在全部八组配对上进行训练，表 1 中的定量结果表示这八次实验的平均性能。

模型	场景	峰值信噪比↑	结构相似性指数↑	学习感知图像块相似度↓
Instant-NGP (基线)	$\mathcal{S}$	26.508	**0.913**	**0.078**
	$\mathcal{B}$	32.436	0.985	0.014
StegoNGP (我们的方法)	$\mathcal{S}$	**26.581**	0.910	0.083
	$\mathcal{B}$	**34.048**	**0.988**	**0.007**

表1. Blender 合成数据集上 StegoNGP（我们的方法）与 Instant-NGP（基线）的重建质量（峰值信噪比、结构相似性指数和学习感知图像块相似度）定量比较。基线方法独立重建每个场景，而我们的方法 StegoNGP 则重建载体场景 $\mathcal{S}$，其中嵌入了一个隐藏场景 $\mathcal{B}$。

如表1所示，我们的 StegoNGP 成功以高保真度重建了载体场景 $\mathcal{S}$ 和隐藏场景 $\mathcal{B}$。定量指标与 Instant-NGP 基线高度相当（在某些情况下略有超越）。这表明我们的联合训练过程不会导致显著的质量下降。在该数据集中观察到的轻微提升可能归因于正则化效应。

图3 显示，StegoNGP 渲染的图像与基线渲染结果高度一致。这是实现隐写不可感知性的关键要求，因为载体场景$\mathcal{S}$不会出现任何可将其与标准 Instant-NGP 模型区分开的视觉伪影。如图3所示，使用无关密钥 $\mathcal{O}$生成的无规律噪声展示了模型的密码学安全性。

4.3. 密钥敏感性与合理否认性

除了重建保真度之外，Ste-goNGP 的核心贡献还取决于其密码学安全性。在本节中，我们分析模型抵御密钥攻击的稳健性，以及其实现可信否认的能力。

为了验证所提出模型的有效性，我们使用训练后的 StegoNGP（乐高作为 $\mathcal{S}$，麦克风作为 $\mathcal{B}$），并向其输入四种不同类型的密钥：默认密钥 Π、正确的秘密密钥 $\mathcal{K}$、部分正确的（混合）密钥（其中一些素数来自 Π，另一些来自 $\mathcal{K}$），以及完全错误的密钥 $\mathcal{O} \equiv \{o_1, o_2, o_3\}$。结果见图4。

如图4所示，StegoNGP 展现了两项关键安全特性。首先，使用默认密钥 Π时，模型能够完美渲染载体场景 $\mathcal{S}$，其外观与标准的非隐写模型完全相同，因此具备合理否认性。其次，该模型表现出很高的密钥敏感性。只有使用准确的秘密密钥 $\mathcal{K}$才能恢复隐藏场景 $\mathcal{B}$。使用部分正确（混合）密钥或其他错误密钥进行尝试均会失败，并产生无规律噪声。这证实了安全性依赖于完整的秘密密钥，因此能够有效抵御暴力破解或部分密钥发现攻击。

4.4. 与现有方法的比较

我们将 StegoNGP 与 GS-Hider [23]，进行比较。GS-Hider 是一种基于 3DGS 的现有隐写术方法。为满足 GS-Hider 对配对相机姿态的要求（载体场景与隐藏场景必须共享相同姿态），我们专门为此次比较生成了一个 Blender 数据集。我们使用对齐的相机矩阵重新合成了全部 8 个场景（每个场景分别包含 100 个训练视图、100 个测试视图和 100 个验证视图），并创建了四组场景对（热狗-乐高、麦克风-Drums、椅子-船和榕树-材料）。下文展示的实验结果为这四组场景对的平均值。与我们采用的密码学零开销方法不同，GS-Hider 会改变模型架构，并依赖额外的神经网络解码器来分离载体场景和隐藏场景。我们从重建保真度、模型尺寸开销和结构可检测性三个方面分析这两种策略之间的权衡。

如表 2所示，尽管 GS-Hider 在场景重建保真度方面表现领先，但这是以显著增加模型开销为代价的。该方法平均引入了 20.54MB 的额外参数，并改变了模型架构，因此在结构上可被检测到。即使不使用复杂的分类器，外部观察者也能仅通过检查文件大小或参数量，准确识别出 GS-Hider 模型。这显然违背了隐写术的根本目标。相比之下，我们的 StegoNGP 实现了零参数开销，无论在架构还是大小上，都与标准的 Instant-NGP 模型无法区分。

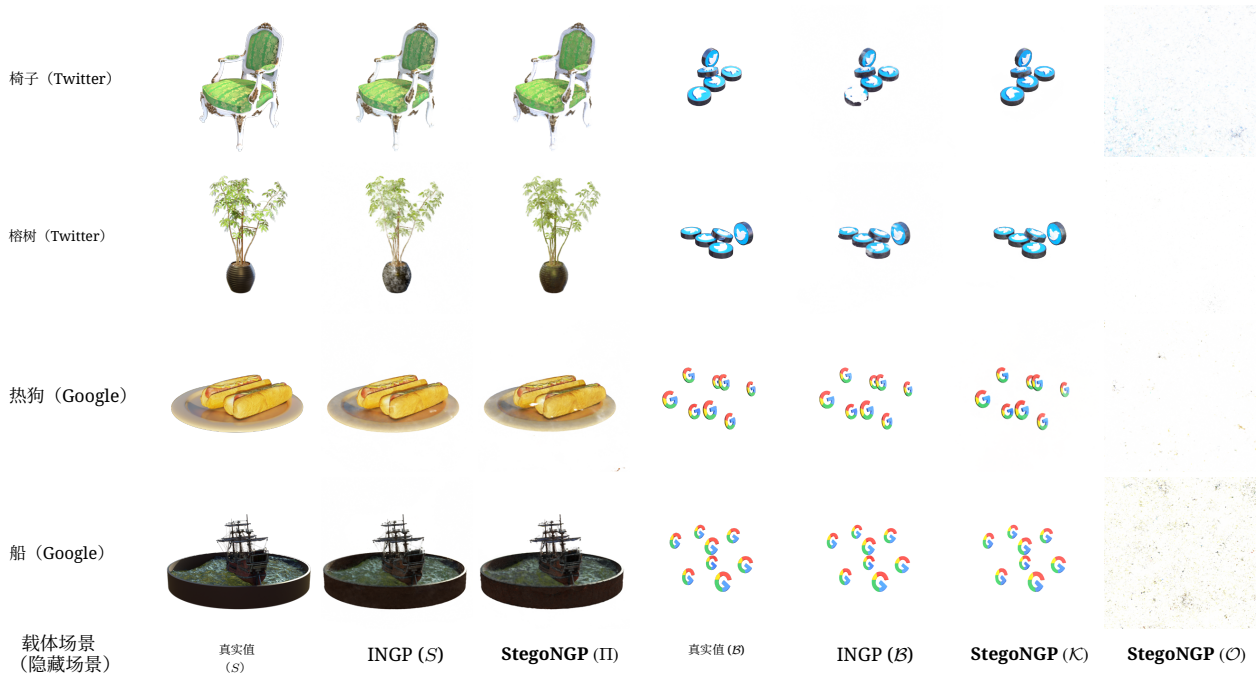

图3. 对比载体场景 $\mathcal{S}$ 和隐藏场景 $\mathcal{B}$ 的真实值（GT）图像，以及基线 Instant-NGP（INGP）和我们提出的 StegoNGP 的渲染结果。我们的模型使用默认密钥 Π渲染场景 $\mathcal{S}$，并使用秘密密钥 $\mathcal{K}$揭示场景 $\mathcal{B}$。图中还展示了使用无关密钥 $\mathcal{O}$得到的空白或噪声输出。

场景 模型	载体场景 $\mathcal{S}$ 峰值信噪比↑	结构相似性指数↑	学习感知图像块相似度↓	隐藏场景 $\mathcal{B}$ 峰值信噪比↑	结构相似性指数↑	学习感知图像块相似度↓	附加参数 模型大小 ↓	结构 可检测性
GS-Hider [23]	**36.194**	**0.991**	**0.006**	**29.625**	**0.962**	**0.030**	+ 20.54 MB	+
StegoNGP（我们的方法）	29.678	0.946	0.056	26.248	0.897	0.094	**+ 0 MB**	-

表2. 对比 StegoNGP（我们的方法）与 GS-Hider [23] 方法在重建保真度（峰值信噪比、结构相似性指数和学习感知图像块相似度）、额外模型大小（MB）和结构可检测性方面的表现。

4.5. 真实世界场景中的性能

为评估 StegoNGP 在复杂真实世界数据上的泛化能力，我们在具有挑战性的 Mip-Nerf-360 数据集上进行了实验。该数据集包含具有复杂光照的无界大型场景。我们展示了 StegoNGP 的性能可视化结果，如图 6所示，并将渲染结果与真实值进行比较。

如图 6所示，StegoNGP 成功学会了同时表示复杂的载体场景和嵌入的隐藏场景。这证明了我们这种高容量加密隐写术方法不仅适用于合成数据集，也适用于真实世界场景。

4.6. 多密钥方案消融研究

在本小节中，我们分析第 3.4 节介绍的多密钥 StegoNGP 方案的有效性。该方案的一个重要动机（其中 $m > 1$）是显著提高相较于标准（$m = 1$）配置的密码学安全性。我们通过模拟外部观察者仅持有不完整秘密密钥集的情形，研究这种多密钥方法的稳健性。在该实验中，向模型提供正确密钥集合 $\mathbf{K}$ 中不同比例的密钥，其余密钥槽则填入默认公钥 Π。这用于评估多密钥方案对部分密钥泄露攻击的抵御能力及其安全阈值。

从图5可以看出，结果表明，多密钥方案具有较高的安全阈值。在0%密钥提供量下，模型会正确地默认渲染载体场景 $\mathcal{S}$，从而实现可信的否认性。即使掌握了大量部分密钥信息（$e.g.$，50%），模型仍无法重建隐藏场景，反而会生成不连贯的输出。虽然在密钥提供量较高时（$e.g.$，75%）可能隐约显现隐藏场景的某些特征，但只有提供完整的密钥集（100%）才能实现连贯且可用的重建

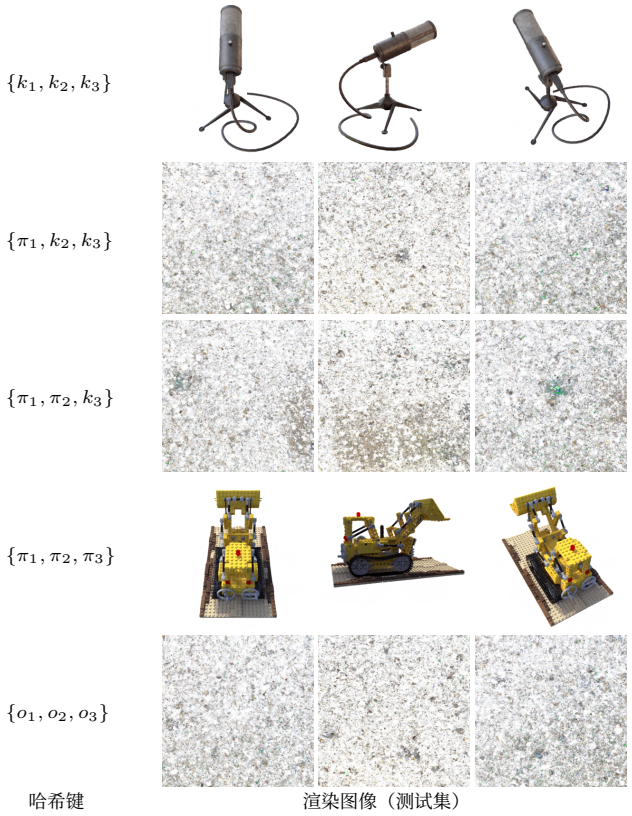

图4. StegoNGP 中密钥敏感性和安全性的演示。只有提供精确的默认密钥 $\{\pi_1, \pi_2, \pi_3\}$ 或秘密密钥 $\{k_1, k_2, k_3\}$ 时，模型才会渲染载体场景（乐高）和隐藏场景（麦克风）。部分正确（混合）的密钥或其他错误密钥均无法解码出任何连贯信息，最终产生噪声。



图 5. 多密钥 StegoNGP 方案及其对部分密钥泄露的鲁棒性。各列表示不同的密钥总数（$m = 4$、8、16）。各行展示了正确密钥集合 **K**所占比例从 0% 增至 100% 时的渲染结果（缺失的密钥由默认密钥 Π替代）。模型从载体场景 $\mathcal{S}$（提供 0% **K**）过渡到隐藏场景 $\mathcal{B}$（提供 100% **K**）。

。这项分析表明，多密钥方案能够有效保护隐藏场景 $\mathcal{B}$，因为攻击者必须掌握完整的 m 密钥集才能成功恢复。

StegoNGP 的关键方案大幅扩展了密钥空间。消融研究证实，该方案能够高度抵抗部分密钥泄露，确保只有使用完整且正确的密钥集才能访问隐藏场景。

5. 结论

本文介绍了一种使用 Instant-NGP 的新型 3D密码学隐写方法（StegoNGP），旨在解决零参数开销下高容量数据隐藏的挑战。我们的核心贡献是将 Instant-NGP 哈希函数转换为由密钥控制的场景切换器，使单个模型能够使用不同的密码密钥同时嵌入公开载体场景和秘密隐藏场景。实验结果表明，我们的方法具有出色的不可感知性，渲染出的载体场景与标准基线难以区分，而其他方法在结构上仍可被检测。此外，我们还提出了一种增强的多



图 6. StegoNGP 在 Mip-Nerf-360 数据集上的性能可视化。该图将多个场景对的渲染结果与各自的真实值图像进行了比较。场景对以载体场景（隐藏场景）的形式表示。

参考文献

[1] Jonathan T Barron、Ben Mildenhall、Matthew Tancik、Peter Hedman、Ricardo Martin-Brualla 和 Pratul P Srinivasan。Mip-NeRF：一种用于抗锯齿神经辐射场的多尺度表示。载于 IEEE/CVF 国际计算机视觉会议论文集，第 5855–5864 页，2021年。2。[2] Jonathan T Barron、Ben Mildenhall、Dor Verbin、Pratul P Srinivasan 和 Peter Hedman。Mip-NeRF 360：无界抗锯齿神经辐射场。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 5470–5479 页，2022年。2，5[3] Zixuan Chen、Guangcong Wang、Jiahao Zhu、Jianhuang Lai 和 Xiaohua Xie。Guardsplat：面向 3D 高斯泼溅的高效且稳健的水印技术。载于计算机视觉与模式识别会议论文集，第 16325–16335 页，2025年。2[4] Weina Dong、Jia Liu、Yan Ke、Lifeng Chen、Wenquan Sun 和 Xiaozhong Pan。通过后门植入对神经辐射场进行隐写。arXiv 预印本 arXiv:2309 10503，2023年。2，3。[5] Danny Driess、Ingmar Schubert、Pete Florence、Yunzhu Li 和 Marc Toussaint。利用神经辐射场进行强化学习。载于《神经信息处理系统进展》，第 16931–16945 页。Curran Associates, Inc.，2022年。2[6]Jiaxin Hong、Sixu Chen、Shuoyang Sun、Hongyao Yu、Hao Fang、Yuqi Tan、Bin Chen、Shuhan Qi 和 Jiawei Li。GaussTrap：针对 3D 高斯泼溅的隐蔽投毒攻击，实现定向场景混淆。arXiv 预印本 arXiv:2504.20829，2025年。2，3[7] Youngdong Jang、Dong In Lee、MinHyuk Jang、Jong Wook Kim、Feng Yang 和 Sangpil Kim。Waterf：用于保护版权的辐射场稳健水印。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 12087–12097 页，2024年。2[8] Youngdong Jang、Hyunje Park、Feng Yang、Heeju Ko、Euijin Choo 和 Sangpil Kim。3D-GSW：用于鲁棒水印。在《计算机视觉与模式识别会议论文集》中，页码 5938–5948，2025。2 –。[9] Wenxiang Jiang、Hanwei Zhang、Shuo Zhao、Zhongwen Guo 和 Hao Wang。IPA-NeRF：针对神经辐射场的幻觉投毒攻击。arXiv 预印本 arXiv:2407.11921，2024。2, 3[10] Bernhard Kerbl、Georgios Kopanas、Thomas Leimk¨uhler 和 George Drettakis。用于实时辐射场渲染的 3D 高斯泼溅。《ACM图形学汇刊》，42(1)，2023，139–1。2, 3[11] Chenxin Li、Brandon Y Feng、Zhiwen Fan、Panwang Pan 和 Zhangyang Wang。StegaNeRF：在神经辐射场中嵌入不可见信息。载于《IEEE/CVF国际计算机视觉会议论文集》，第 441–453 页，2023。2, 3[12] Lijiang Li、Jinglu Wang、Xiang Ming 和 Yan Lu。GS-marker：适用于 3D 高斯泼溅的通用鲁棒水印。arXiv 预印本 arXiv:2503.18718，2025。2[13] David B Lindell、Julien NP Martel 和 Gordon Wetzstein。AutoInt：用于快速神经体渲染的自动积分。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 14556–14565 页，2021。2[14] Ben Mildenhall、Pratul P Srinivasan、Matthew Tancik、Jonathan T Barron、Ravi Ramamoorthi 和 Ren Ng。NeRF：将场景表示为用于视图合成的神经辐射场。《ACM通讯》，65(1):99–106，2021。1, 2, 5[15] Thomas M¨uller、Alex Evans、Christoph Schied 和 Alexander Keller。采用多分辨率哈希编码的即时神经图形基元。《ACM图形学汇刊（TOG）》，41(4):1–15，2022。1, 2, 5[16] Albert Pumarola、Enric Corona、Gerard Pons-Moll 和 Francesc Moreno-Noguer。D-NeRF：用于动态场景的神经辐射场。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 10318–10327 页，2021。2

[17] Daniel Rebain、Wei Jiang、Soroosh Yazdani、Ke Li、Kwang Moo Yi 和 Andrea Tagliasacchi。Derf：分解式辐射场。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 14153–14161 页，2021。2[18] Qi Song、Ziyuan Luo、Ka Chun Cheung、Simon See 和 Renjie Wan。通过即插即用水印基础模型保护 NeRF 的版权。载于《欧洲计算机视觉会议论文集》，第 57–73 页。施普林格，2024。2[19] Wenquan Sun、Jia Liu、Weina Dong、Lifeng Chen 和 Fuqiang Di。RWNerf：基于可逆神经网络的神经辐射场鲁棒水印方案。《计算机、材料与连续体》，80(3)，2024。2[20] Matthew Tancik、Vincent Casser、Xinchen Yan、Sabeek Pradhan、Ben Mildenhall、Pratul P Srinivasan、Jonathan T Barron 和 Henrik Kretzschmar。Block-NeRF：可扩展的大场景神经视图合成。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 8248–8258 页，2022。2[21] Marie A Wright。高级加密标准。《网络安全》，2001年(10):11–13，2001年。2[22] Guanjun Wu、Taoran Yi、Jiemin Fang、Lingxi Xie、Xiaopeng Zhang、Wei Wei、Wenyu Liu、Qi Tian 和 Xinggang Wang。用于实时动态场景渲染的 4D高斯泼溅。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 20310–20320 页，2024。2[23] Xuanyu Zhang、Jiarui Meng、Runyi Li、Zhipei Xu、Yongbing Zhang 和 Jian Zhang。GS-Hider：将消息隐藏于 3D 高斯泼溅中。《神经信息处理系统进展》，37:49780–49805，2024。2, 3, 5, 6, 7

StegoNGP：使用 Instant-NGP 的 3D密码学隐写

补充材料

本补充文档提供了额外细节、定量分析和消融研究，以支持主论文中提出的主张。我们首先给出详细的安全性分析，对密码密钥空间以及抵御暴力攻击的稳健性进行定量分解。随后，我们提供哈希冲突与表大小分析，研究哈希表容量对复杂场景中重建保真度的影响。在此之后，我们给出主论文中所总结的全部实验的完整逐场景详细实验结果，包括合成数据集、真实世界数据集、与现有方法的比较，以及多密钥消融研究的详细指标。最后，我们纳入一项关于嵌入多个隐藏场景的附加实验，以探索本方法的容量极限。

6. 安全性分析

本节对 StegoNGP 方案的密钥空间及其抵御暴力攻击的稳健性进行定量分析。我们假设攻击者拥有完整的模型和算法知识，但不掌握密钥（$\mathcal{K}$ 或 $\mathbf{K}$），因而必须穷举搜索所有密钥组合。

6.1. 密钥空间分析

我们的**基础 StegoNGP** 方法（$m = 1$）采用从范围 $[10^7, 10^{10}]$内的素数池中选取的三素数元组（$\mathcal{K} \equiv \{k_1, k_2, k_3\}$）。设 $P(x)$ 为范围 $(0, x)$ 内素数个数的素数计数函数，则该范围内可用素数的总数 U 为：

$$\begin{aligned} U &= P(10^{10}) - P(10^7) \\ &= 455,052,511 - 664,579 = 454,387,932, \end{aligned} \tag{11}$$

总密钥空间 Q_{basic} 及其对应的比特强度为：

$$Q_{basic} = U^3 = (454,387,932)^3 \approx 9.364 \times 10^{25}, \tag{12}$$

$$\text{Bit Strength} = \log_2(Q_{basic}) \approx \mathbf{86.3} \text{ bits}, \tag{13}$$

2^{86} 的密钥空间在计算上是安全的。对于**多密钥 StegoNGP**（$m = 16$），攻击者必须找出全部 16×3 个素数密钥（总计 48 个素数）。总密钥空间 Q_{multi} 为：

$$\begin{aligned} Q_{multi} &= (Q_{basic})^{16} = (U^3)^{16}, \\ \text{Bit Strength} &= \log_2(Q_{multi}) \approx 16 \times (86.3) \approx \mathbf{1380} \text{ bits}, \end{aligned} \tag{14}$$

其密钥空间远大于 AES-256 等加密标准，面对当前的计算能力，破解它在计算上不可行。

6.2. 暴力破解可行性

我们在对攻击者极为有利的假设下，分析破解 **基础模型**($Q_{basic} \approx 9.36 \times 10^{25}$) 的可行性：(1) 使用具有充足显存的高性能图形处理器，例如 NVIDIA RTX 5090（32GB 显存和每秒 3,352 万亿次运算的计算能力）；(2) 渲染速度为每秒 20 张图像（20 帧/秒）；(3) 使用理想且即时的分类器来区分连贯场景与噪声。

假设搜索密钥空间的 50%，破解（T_{years}）的平均时间为：

$$T_{\text{sec}} = \frac{Q_{basic} \times 0.5}{20} \approx 2.341 \times 10^{24} \text{ seconds} \tag{15}$$

$$T_{\text{years}} = \frac{2.341 \times 10^{24} \text{ sec}}{3.15 \times 10^7 \text{ sec/year}} \approx 7.43 \times 10^{16} \text{ years} \tag{16}$$

即使在这些理想假设下，要破解我们的基本模型也需要大约 **7430 万亿年**，这在计算上不可行。我们的定量分析汇总于表 3，证实了 StegoNGP 方案在密码学上的稳健性。

7. 哈希冲突与表大小分析

在实验中，我们观察到，简单的合成数据集与复杂、大规模的真实世界数据集（如 Mip-Nerf-360）之间存在性能差距（详细结果见表 8）。我们推测，这种性能下降主要是由于 Instant-NGP 多分辨率哈希编码的特征表中哈希冲突率上升所致。

7.1. 冲突分析

Instant-NGP 通过将稀疏高分辨率网格中的坐标映射到固定大小的密集特征表 T，实现了高效性。正如 Instant-NGP 论文所述，3D 空间是稀疏的；例如，一个分辨率为 128^3 的网格（包含超过 200 万个单元）可能只有约 54，000 个活跃单元实际参与构成可见表面。

在标准的 Instant-NGP 中，这些 $\kappa_1 \approx 54{,}000$ 个活跃单元通过哈希函数映射到特征表 T 中。我们的模型采用默认密钥表大小 $T = 2^{19}$ =524,288。理想情况下，这会产生 $\rho_1 = \kappa_1/T \approx 54,000/524,288 \approx 10.3$ % 的填充率。虽然碰撞不可避免，但稀疏性和较大的表大小使其在单个场景中保持在可控范围内。因此，原始 Instant-NGP 论文 [15] 并未明确处理哈希冲突。

然而，我们的 StegoNGP 方法将两个完整的 3D 场景（载体 S 和隐藏场景 $\mathcal{B}$）交织到同一个 特征

表3. StegoNGP 密钥空间与安全性摘要

模型	密钥组成	密钥空间大小	比特强度	估计暴力破解时间（RTX 5090 @ 20 帧/秒）
基础($m=1$)	3个素数	$\approx 9.36\times10^{25}$	$\approx 2^{86.3}$	$\approx 7.43\times10^{16}$ 年
多密钥($m=16$)	48个素数	$\approx 3.46\times10^{415}$	$\approx 2^{1380}$	计算上不可行
AES-256 [21]	256 位密钥	$\approx 1.15\times10^{77}$	2^{256}	计算上不可行

表 T中。载体场景 S使用默认密钥 Π，将其 κ_1 活跃单元映射到 T。与此同时，隐藏场景 $\mathcal{B}$使用秘密密钥 $\mathcal{K}$，将其 κ_2 活跃单元映射到相同的 T。

假设隐藏场景具有相似的复杂度（例如，$\kappa_2 \approx 54,000$），映射到 T的活跃单元总数现在为 $\kappa_{total} = \kappa_1 + \kappa_2 \approx 108,000$。这实际上将哈希表的填充率翻了一倍，达到 $\rho_{stego} = \kappa_{total}/T$ ≈20。6%。这显著增加了哈希冲突的概率，即来自场景 S 和场景 $\mathcal{B}$ 的特征向量被哈希到同一个表项，从而在训练期间造成特征干扰。这种干扰迫使网络对该槽位的特征向量做出次优折中，降低两个场景的重建质量。

7.2. 实验验证

为了验证这一假设，我们设计了一项实验，用于测量哈希表大小 T 对复杂真实场景对重建保真度的影响。我们采用了 Mip-Nerf-360 数据集中的“盆景”（覆盖场景 S）和“厨房”（隐藏场景 $\mathcal{B}$）这一场景对，并使用三种不同的哈希表大小训练了 StegoNGP 模型：

- $T = 2^{18}$ （一半：262,144 个条目）；
- $T = 2^{19}$ （默认密钥：524,288 个条目）；
- $T = 2^{20}$ （两倍：1,048,576 个条目）；
- $T = 2^{21}$ （四倍：2,097,152 个条目）。

结果见表4。

表 4. 在 Mip-Nerf-360 “盆景”（S）/“厨房”（$\mathcal{B}$）场景对上改变哈希表大小 T的定量结果。随着 T增大，哈希冲突概率降低，使封面场景和隐藏场景的重建保真度均明显提升。

表大小（T）	场景	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓
$T = 2^{18}$	S	23.145	0.715	0.317
	$\mathcal{B}$	22.797	0.572	0.393
$T = 2^{19}$	S	22.862	0.721	0.286
	$\mathcal{B}$	22.888	0.575	0.398
$T = 2^{20}$	S	23.626	0.754	0.235
	$\mathcal{B}$	22.695	0.579	0.388
$T = 2^{21}$	S	22.998	0.753	0.228
	$\mathcal{B}$	22.941	0.621	0.343

表 4 中的结果有力地支持了我们的假设。通过将 $T = 2^{20}$加倍以及将$T = 2^{21}$扩大至四倍，我们为两个场景的特征向量提供了更多空间，显著降低了碰撞和特征干扰的发生率。这使得覆盖场景和隐藏场景的 PSNR、SSIM 和 LPIPS 均得到持续且明显的提升。相反，当我们将哈希表缩小一半（$T = 2^{18}$）时，重建质量会下降。

8. 详细实验结果

本节详细分析了主论文中所呈现实验的定量指标，包括合成数据集的完整结果（第4.2节）、与现有方法的逐对比较（第4.4节）、真实世界数据集的定量指标（第4.5节），以及多密钥消融研究的详细数据（第4.6节）。

8.1. 合成场景上的详细性能（第4.2节）

表5 对主论文中的表1 作了进一步补充。表中展示了 Blender 合成数据集实验所用全部八组场景对各自的重建保真度（峰值信噪比、结构相似性指数、学习感知图像块相似度）。平均值行对应于主论文表*1*中的StegoNGP（我们的方法）行。

8.2. 与现有方法的详细比较（第4.4节）

表6和7 给出了主论文表2 所总结比较结果的逐场景对明细。该实验采用了四组相机姿态对齐的场景对，具体如第4.4节所述。表6 展示了 GS-Hider 的结果 [23], , 表7 展示了我们提出的 StegoNGP 的结果。

8.3. 真实世界场景上的详细性能（第4.5节）

表8 给出了 Mip-Nerf-360 数据集实验的定量指标（峰值信噪比、结构相似性指数、学习感知图像块相似度）[2]。这些指标对应于主论文图6 所示的定性可视化结果。

8.4. 多密钥方案的详细消融研究（第4.6节）

本节提供主论文第 4.6 节消融研究的详细数据。我们首先

表 5. Blender 合成数据集上的详细定量结果。每一行代表一个经过训练、将一个场景隐藏在另一个场景中的 StegoNGP 模型。这些指标展示了渲染载体场景 S（使用密钥 Π）和隐藏场景 $\mathcal{B}$（使用密钥 $\mathcal{K}$）的保真度。

场景对		载体场景（S）			隐藏场景（$\mathcal{B}$）		
载体场景（S）	隐藏场景（$\mathcal{B}$）	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓
Lego	Twitter	26.902	0.910	0.070	32.514	0.985	0.009
Mic	Twitter	28.361	0.955	0.052	32.693	0.984	0.010
椅子	Twitter	28.308	0.921	0.086	32.630	0.986	0.009
榕树	Twitter	26.950	0.935	0.044	32.891	0.985	0.009
热狗	Google	29.586	0.946	0.056	34.991	0.991	0.006
Drums	Google	22.730	0.894	0.103	35.804	0.992	0.005
Ship	Google	24.847	0.814	0.181	34.902	0.991	0.006
材料	Google	24.965	0.903	0.070	35.957	0.992	0.005
平均值		**26.581**	**0.910**	**0.083**	**34.048**	**0.988**	**0.007**

表 6. **GS-Hider [23]** 方法在四个专用场景对上的详细结果。这些指标的平均值对应主论文表2 中的 GS-Hider 行。

场景对		载体场景 (S)			隐藏场景 ($\mathcal{B}$)		
载体场景（S）	隐藏场景（$\mathcal{B}$）	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓
热狗	Lego	35.376	0.985	0.009	30.048	0.971	0.016
Mic	Drums	36.601	0.993	0.005	27.091	0.959	0.028
椅子	Ship	37.055	0.992	0.007	29.460	0.931	0.027
榕树	材料	35.744	0.995	0.004	31.899	0.985	0.049
平均值		**36.194**	**0.991**	**0.006**	**29.625**	**0.962**	**0.030**

表 7。我们 **StegoNGP（我们的方法）**方法在四个专用场景对上的详细定量结果。这些指标的平均值对应于主论文表 2 中的 StegoNGP（我们的方法）行。

场景对		载体场景（S）			隐藏场景（$\mathcal{B}$）		
载体场景（S）	隐藏场景（$\mathcal{B}$）	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓
热狗	Lego	32.269	0.953	0.052	28.100	0.923	0.062
Mic	Drums	29.213	0.957	0.050	24.017	0.907	0.093
椅子	Ship	28.796	0.925	0.085	26.184	0.832	0.166
榕树	材料	28.434	0.948	0.036	26.689	0.926	0.056
平均值		**29.678**	**0.946**	**0.056**	**26.248**	**0.897**	**0.094**

表 8。StegoNGP 在 Mip-Nerf-360 数据集上的详细结果。每一行对应图中可视化的一个场景对6。

场景对		载体场景 (S)			隐藏场景 ($\mathcal{B}$)		
载体场景（S）	隐藏场景（$\mathcal{B}$）	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓
盆景	厨房	22.862	0.721	0.286	22.888	0.575	0.398
花园	Treehill	20.559	0.405	0.578	16.951	0.430	0.545
厨房	台面	21.289	0.550	0.425	21.854	0.678	0.397
Room	盆景	24.696	0.789	0.259	21.995	0.707	0.264
平均值		**22.352**	**0.616**	**0.387**	**20.922**	**0.598**	**0.401**

列出用于多密钥实验的 16 个秘密密钥集合（表9），其中前 4 个密钥用于$m = 4$，前 8 个密钥用于 $m = 8$，全部 16 个密钥用于$m = 16$。随后，我们提供支持图5中部分密钥泄露分析的定量指标（表10），展示随着正确秘密密钥比例的增加，重建保真度的变化。

表 9。多密钥 StegoNGP 实验中使用的 16 个秘密密钥主集合。每个密钥 $\mathcal{K}_j$ 是一个三素数元组 $\{k_{j,1}, k_{j,2}, k_{j,3}\}$。

密钥 ID (j)	素数 $k_{j,1}$	素数 $k_{j,2}$	素数 $k_{j,3}$
1	10123105283	9423208789	10600480711
2	3674653429	2097192037	1434869437
3	8542170799	6143207549	6519670961
4	15443898139	12668029511	12301111429
5	14405672371	5760082793	11538455387
6	13705036483	5104709759	3761188213
7	3364878689	5035457381	1560686051
8	10808744237	9293063341	12211915819
9	9320630941	2357489971	12349316797
10	5123446871	10280445341	8301841781
11	5489304827	13394370001	11743146431
12	13560374447	8074960889	6193177019
13	3510659183	2993894131	8230892477
14	5512642787	4108776493	15236738411
15	8923388921	1689310319	8101897283
16	8288642353	3761544863	7037980441

9. 嵌入多个隐藏场景

为了进一步探索 StegoNGP 的容量上限，我们设计了一项实验，将多个隐藏场景嵌入到单个载体场景模型中。主论文展示了 $\Phi = 1$ 的情况，即隐藏一个完整的 3D 场景。在这里，我们将其扩展到 $\Phi = 2$、3 和 4 个隐藏场景。

这是通过扩展算法 1 中的训练逻辑实现的。我们不再随机选择 S（使用密钥 Π）或 $\mathcal{B}$（使用密钥 $\mathcal{K}$），而是从集合 $\{S, \mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_\Phi\}$ 中随机采样，其中每个场景 $\mathcal{B}_i$ 都分配有唯一的秘密密钥 $\mathcal{K}_i$。

我们假设，随着 Φ 增加，所有场景的重建质量都会下降。这是第 7 节所分析的哈希碰撞问题的直接结果。对于一个载体场景和 Φ 个隐藏场景，被映射到同一个特征表 T 中的活跃单元总数为 $\kappa_{total} \approx (\Phi + 1) \times \kappa$，这会显著加剧特征干扰。

我们使用 Blender 合成数据集进行此测试，并采用 $T = 2^{19}$ 默认哈希表大小。表11中的结果证实了我们的假设。

表 10。Multi-Key StegoNGP 部分密钥泄露攻击的详细结果，对应主论文中的图5（使用热狗作为载体场景 S，椅子作为隐藏场景 $\mathcal{B}$）。我们测量正确秘密密钥（$\mathbf{K}$）比例增加时的重建保真度。

模型	K 提供量（%）	峰值信噪比 (S)	峰值信噪比 $(\mathcal{B})$
$m = 4$	0%（渲染 S）	**30.620**	11.805
	25%	15.860	13.761
	50%	13.822	16.219
	75%	12.640	19.613
	100%（渲染 $\mathcal{B}$）	11.876	**27.332**
$m = 8$	0%（渲染 S）	**30.093**	11.680
	25%	13.421	14.255
	50%	12.909	15.746
	75%	12.243	19.548
	100%（渲染 $\mathcal{B}$）	11.857	**27.513**
$m = 16$	0%（渲染 S）	**30.556**	11.850
	25%	14.551	14.775
	50%	12.676	18.193
	75%	12.144	21.404
	100%（渲染 $\mathcal{B}$）	11.877	**27.731**

表 11. 使用默认 $T = 2^{19}$ 哈希表将 Φ个隐藏场景嵌入单个模型的定量结果。随着 Φ 增加，哈希冲突增多，导致所有场景的重建保真度明显下降。

Φ	场景	峰值信噪比 ↑	结构相似性指数 ↑	学习感知图像块相似度 ↓
1	S (热狗)	30.726	0.949	0.053
	$\mathcal{B}_1$ (椅子)	27.508	0.919	0.090
2	S (热狗)	31.606	0.950	0.055
	$\mathcal{B}_1$ (椅子)	28.141	0.919	0.090
	$\mathcal{B}_2$ (麦克风)	28.566	0.955	0.051
3	S (热狗)	31.037	0.946	0.063
	$\mathcal{B}_1$ (椅子)	27.699	0.914	0.094
	$\mathcal{B}_2$ (麦克风)	28.137	0.953	0.056
	$\mathcal{B}_3$ (乐高)	26.496	0.900	0.083
4	S (热狗)	29.987	0.940	0.069
	$\mathcal{B}_1$ (椅子)	27.331	0.907	0.105
	$\mathcal{B}_2$ (麦克风)	27.497	0.949	0.061
	$\mathcal{B}_3$ (乐高)	25.798	0.887	0.094
	$\mathcal{B}_4$ (船)	24.663	0.800	0.204



图7. imag的可视化 同时嵌入多个（Φ）隐藏场景（$\mathcal{B}_1$、 $\mathcal{B}_2$、 $\mathcal{B}_3$ 或 $\mathcal{B}_4$）并覆盖所得的结果 r
场景（S）到单个模型中 使用默认密钥 $T = 2^{19}$ 哈希表。

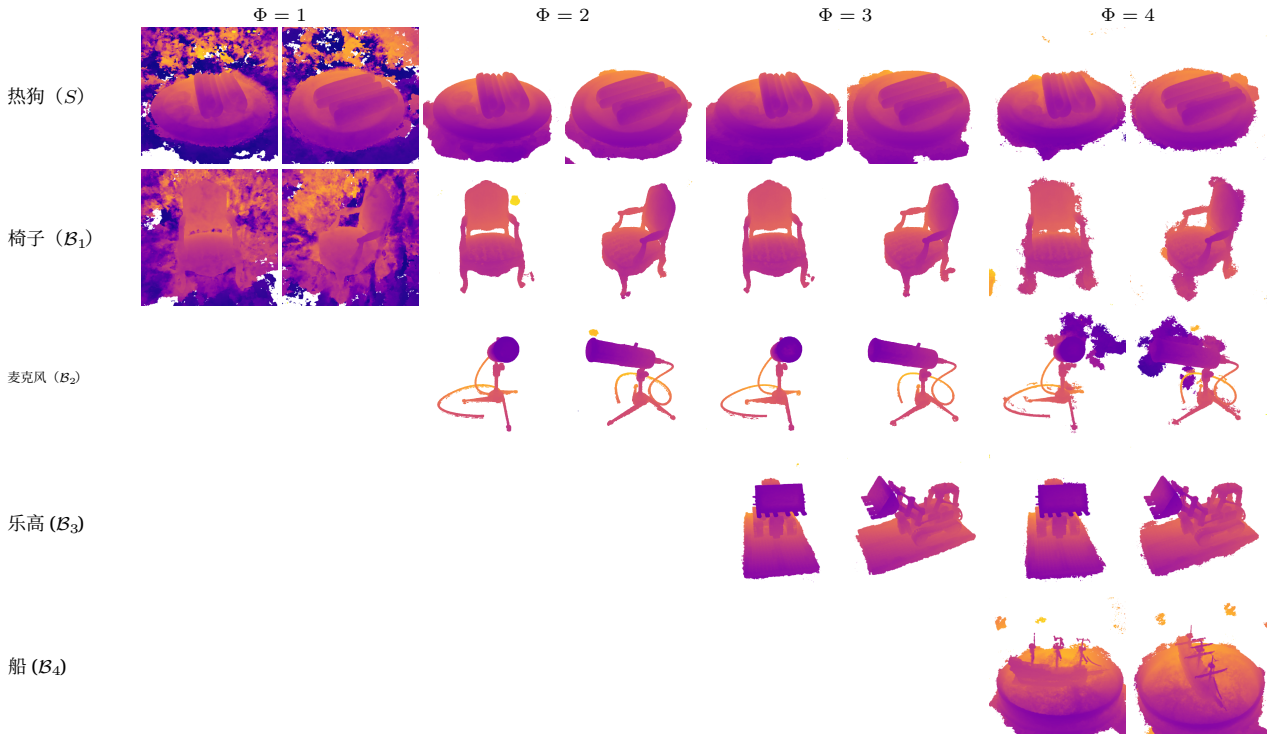

图8. 使用默认 $T = 2^{19}$ 哈希表将多个（Φ）隐藏场景（$\mathcal{B}_1$、 $\mathcal{B}_2$、 $\mathcal{B}_3$ 或 $\mathcal{B}_4$）与封面场景（S）同时嵌入单个模型时的深度结果可视化。