RegGS: Unposed Sparse Views Gaussian Splatting with 3DGS Registration

Supplementary Material

6. Entropy-Regularized Sinkhorn $W2$ Distance Gradient Consistency Proof

The entropy-regularized Wasserstein distance $W_{2,\epsilon}^2(G_A, T(G_B))$, where $\epsilon > 0$ is a regularization parameter, provides a computationally feasible approach to the infinite-dimensional optimization problem inherent in calculating the exact Wasserstein distance W_2^2. As $\epsilon \to 0$, the gradient $\nabla_\xi W_{2,\epsilon}^2$ converges to the subgradient set of the exact Wasserstein distance, denoted as $\partial W_2^2(\xi)$:

$$\lim_{\epsilon \to 0} \nabla_\xi W_{2,\epsilon}^2 \in \partial W_2^2(\xi). \tag{19}$$

Foundation of Γ-Convergence. According to optimal transport theory [15], the entropy-regularized Wasserstein distance satisfies Γ-convergence: for any probability distributions G_A and G_B, as the regularization parameter ϵ approaches zero,

$$W_{2,\epsilon}^2(G_A, G_B) \xrightarrow{\Gamma} W_2^2(G_A, G_B), \tag{20}$$

where Γ-convergence ensures that the sequence of minima of the regularized problem converges to the optimum of the original problem. Specifically, for a parametrized transformation $T(\xi)$, the minimization of the entropy-regularized objective function $W_{2,\epsilon}^2$ approximates the non-regularized objective W_2^2 in the limit.

6.1. Gradient Expression Derivation.

The entropy-regularized Wasserstein distance is defined as:

$$W_{2,\epsilon}^2 = \min_{\pi \in \Pi(w_A, w_B)} \sum_{i,k} \pi_{ik} C_{ik}(\xi) + \epsilon \sum_{i,k} \pi_{ik} \log \pi_{ik}, \tag{21}$$

where $C_{ik}(\xi) = \|\mu_i^A - \mu_k^{B\prime}\|^2 + \mathrm{Tr}(\Sigma_i^A + \Sigma_k^{B\prime} - 2(\Sigma_i^A \Sigma_k^{B\prime})^{1/2})$ depends on the transformation parameters ξ. Using the implicit function theorem [13], the gradient of the regularized problem can be expressed as:

$$\nabla_\xi W_{2,\epsilon}^2 = \sum_{i,k} \pi_{ik}^*(\epsilon) \nabla_\xi C_{ik}(\xi), \tag{22}$$

where $\pi_{ik}^*(\epsilon)$ is the optimal transport plan under entropy regularization. This expression indicates that the gradient is constituted by a weighted average of the cost function gradients under the transport plan.

6.2. Convergence of the Transport Plan.

As $\epsilon \to 0$, the influence of the entropy regularization term $\epsilon \sum \pi_{ik} \log \pi_{ik}$ diminishes. According to Γ-convergence, any limit point of the regularized transport plan $\pi_{ik}^*(\epsilon)$ is an optimal solution of the original Wasserstein problem, i.e.,

$$\lim_{\epsilon \to 0} \pi_{ik}^*(\epsilon) = \pi_{ik}^* \in \arg\min_{\pi} \sum_{i,k} \pi_{ik} C_{ik}(\xi). \tag{23}$$

Since multiple optimal transport plans may exist (e.g., multiple paths with the same minimum cost), π^* belongs to a set of optimal solutions Π^*.

6.3. Construction of the Subgradient Set

For a nonsmooth convex function W_2^2, its Clarke subgradient is defined as:

$$\partial W_2^2(\xi) = \left\{ \sum_{i,k} \pi_{ik}^* \nabla_\xi C_{ik}(\xi) \mid \pi^* \in \Pi^* \right\}. \tag{24}$$

As $\epsilon \to 0$, the limit points of the regularized gradient $\nabla_\xi W_{2,\epsilon}^2 = \sum_{i,k} \pi_{ik}^*(\epsilon) \nabla_\xi C_{ik}(\xi)$ are determined by the convergence of $\pi_{ik}^*(\epsilon)$. Thus,

$$\lim_{\epsilon \to 0} \nabla_\xi W_{2,\epsilon}^2 = \sum_{i,k} \pi_{ik}^* \nabla_\xi C_{ik}(\xi) \in \partial W_2^2(\xi), \tag{25}$$

indicating that the regularized gradient converges to an element of the subgradient set.

Although W_2^2 may be nonconvex with respect to ξ, it satisfies local Lipschitz continuity on any compact set [32], ensuring the existence of subgradients.

If the original problem has a unique optimal transport plan π^*, then the subgradient reduces to a singleton and the gradient convergence path is unique; otherwise, convergence is toward a specific direction within the subgradient set.

The entropy-regularized gradient $\nabla_\xi W_{2,\epsilon}^2$ asymptotically approaches the exact Wasserstein subgradient direction as ϵ approaches zero. This property theoretically supports the hierarchical optimization strategy of gradually reducing ϵ in our methodology: initially leveraging the smoothness of the regularization term to avoid local minima and eventually converging towards the direction of the exact Wasserstein distance, thus achieving robust global distribution alignment.

7. Sinkhorn Algorithm Complexity

The main map and the submap contain M and N Gaussian gradients, respectively. Initially, the first step of the Sinkhorn algorithm involves constructing a kernel matrix

$K \in \mathbb{R}^{M \times N}$, whose elements are given by

$$K_{ik} = \exp\left(-\frac{C_{ik}}{\epsilon}\right), \tag{26}$$

where C_{ik} represents the 2-Wasserstein cost for the Gaussian pair $(N_i^A,\, N_k^{B'})$. Calculating each C_{ik} includes two parts: one is the distance between means $\|\mu_i^A - \mu_k^{B'}\|^2$, which has a complexity of $O(1)$; the second is the covariance term

$$\mathrm{Tr}\left(\Sigma_i^A + \Sigma_k^{B'} - 2\left(\Sigma_i^A\,\Sigma_k^{B'}\right)^{1/2}\right), \tag{27}$$

where the square root of the covariance matrix is usually implemented via Cholesky decomposition, with a single conjunction complexity of $O(d^3)$ (for a three-dimensional space $d = 3$), but since all Gaussian covariances can be precomputed, this can be considered $O(1)$ in the context of C_{ik} computation. Thus, the overall complexity of constructing the kernel matrix K is $O(MN)$.

Next, during the iteration phase, the Sinkhorn algorithm manages $u \in \mathbb{R}^M$ and $v \in \mathbb{R}^N$ through alternating updates to satisfy the marginal constraints, with the update formulas

$$u^{(t)} = \frac{w_A}{K\, v^{(t-1)}}, \qquad v^{(t)} = \frac{w_B}{K^\top\, u^{(t)}}. \tag{28}$$

Here, the complexity of multiplying the matrix with the support (i.e., computing $K\, v^{(t-1)}$ and $K^\top\, u^{(t)}$) incurs $O(MN)$, while the element-wise division to update $u^{(t)}$ and $v^{(t)}$ has complexities of $O(M)$ and $O(N)$, respectively, which are negligible compared to the previous step. Therefore, each iteration's computational complexity is $O(MN)$.

After T iterations, the total time complexity is the kernel matrix initialization $O(MN)$ plus $T \cdot O(MN)$, which is

$$O(MN) + T \cdot O(MN) = O(TMN). \tag{29}$$

In practice, due to the introduction of the entropy regularization term, which significantly speeds up convergence, according to [15], the Sinkhorn algorithm typically converges within $T \leq 50$ iterations to a relative parameter $\delta < 10^{-3}$, a characteristic that has been verified in multiple optimal transport libraries such as POT [19]. The detailed procedure is described in the following Algorithm 1.

Moreover, although the theoretical time complexity is $O(TMN)$, in engineering implementations, various strategies can reduce the sparsity factor, utilizing GPU sparsity for computing the kernel matrix K and matrix multiplications; employing safe logarithmic domain computations (i.e., computing $\log K_{ik} = -C_{ik}/\epsilon$) to sparsify and reduce multiplication/division operations; and leveraging sparsity methods to speed up by building a sparse kernel matrix reducing the computational complexity to $O(S)$ (where $S \ll MN$ is the number of non-zero elements).

Algorithm 1 Entropy-Regularized Optimal Transport MW_2 Distance

Require:
- Gaussian components: $\{\mathcal{N}(\mu_i^A, \Sigma_i^A)\}_{i=1}^M$ and $\{\mathcal{N}(\mu_k^{B'}, \Sigma_k^{B'})\}_{k=1}^N$.
- Marginal weights: $w^A \in \mathbb{R}^M$ and $w^B \in \mathbb{R}^N$.
- Regularization parameter: $\epsilon > 0$.
- Maximum iterations: T.

Ensure:
- Transport plan $\pi^* \in \mathbb{R}^{M \times N}$.
- Entropy-regularized transport cost $W_{2,\epsilon}^2$.

Step 1: Compute Cost Matrix C
for $i = 1$ to M **do**
  for $k = 1$ to N **do**
    $C_{ik} \leftarrow \|\mu_i^A - \mu_k^{B'}\|^2 + \mathrm{Tr}\Big(\Sigma_i^A + \Sigma_k^{B'} - 2\big(\Sigma_i^A \Sigma_k^{B'}\big)^{1/2}\Big)$
  end for
end for
Step 2: Compute Kernel Matrix K
for $i = 1$ to M **do**
  for $k = 1$ to N **do**
    $K_{ik} \leftarrow \exp\Big(-\frac{C_{ik}}{\epsilon}\Big)$
  end for
end for
Step 3: Initialize scaling vectors
$u \leftarrow \mathbf{1}_M$ ▷ $\mathbf{1}_M$ denotes a vector of ones with length M
$v \leftarrow \mathbf{1}_N$ ▷ $\mathbf{1}_N$ denotes a vector of ones with length N
Step 4: Perform Sinkhorn Iterations
for $t = 1$ to T **do**
  $u \leftarrow w^A \oslash (K\, v)$ ▷ $\oslash$ denotes element-wise division
  $v \leftarrow w^B \oslash (K^\top u)$
end for
Step 5: Compute the Transport Plan
$\pi^* \leftarrow \mathrm{diag}(u)\, K\, \mathrm{diag}(v)$
Step 6: Compute the Entropy-Regularized Transport Cost
$W_{2,\epsilon}^2 \leftarrow \sum_{i=1}^M \sum_{k=1}^N \pi_{ik}^*\, C_{ik}$
return $\pi^*,\ W_{2,\epsilon}^2$

In summary, under entropy regularization, the Sinkhorn algorithm's time complexity is $O(TMN)$, where T is the number of iterations (usually $T \leq 50$). In practical applications of 3D Gaussian map registration (e.g., $M, N \leq 10^5$), a single iteration takes about 10ms, and with GPU sparsification and sparsity priors, this method can achieve fast processing of large-scale 3D Gaussian map registration problems.

8. Additional Experimental Results

Figure 8 illustrates the generalization of our method on real video data, where we uniformly sampled four frames from

Method	2×			16×			64×		
	PSNR↑	Time↓	GPU(GB)↓	PSNR↑	Time↓	GPU(GB)↓	PSNR↑	Time↓	GPU(GB)↓
Splatt3R	13.951	**20s**	7.3	-	-	-	-	-	-
NoPoSplat	23.247	22s	3.9	-	-	-	-	-	-
DUSt3R*	18.484	259s	**3.5**	24.714	**22min**	10.5	OOM	OOM	OOM
MASt3R*	16.036	283s	3.7	24.249	23min	**4.5**	**28.826**	**54min**	41.3
Ours	**24.272**	259s	3.9	**28.663**	57min	12.1	28.703	165min	**12.1**

Figure 7. **Additional quantitative comparison on RE10K showing runtime and memory usage across different input view counts.**

Figure 8. Generalization results using a real video. Four frames were uniformly sampled and used for sparse reconstruction to demonstrate the method's applicability to real-world scenarios.

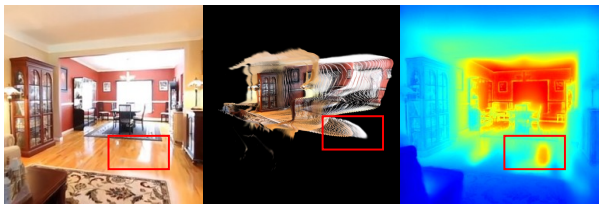

Figure 9. Sub gaussians from NoPoSplat. These sub Gaussians generated by NoPoSplat indicate that, in certain scenes, the Gaussians produced by NoPoSplat exhibit abnormalities in their spatial structure.

a video and used a pretrained feed-forward Gaussian model to extract local 3D Gaussian representations. These representations were registered and fused by the RegGS method into a consistent 3D Gaussian scene. Results confirm the method's effectiveness in achieving precise camera localization and scene alignment even with sparse viewpoints, thus generating high-quality novel views suitable for real-world applications.

We evaluate runtime and memory across sampling ratios from 2x to 64x using 200-frame sequences. As shown in Tab. 7, RegGS maintains controlled memory usage across all input settings. In contrast, NoPoSplat and Splatt3R are limited to two-view inputs, while DUSt3R and MASt3R exhibit exponential growth in Gaussian count, frequently resulting in out-of-memory failures. This demonstrates the scalability of RegGS under sparse view conditions. At 64x, as view coverage becomes denser, the reconstruction bottleneck shifts from view sparsity to the capacity of the 3DGS representation. RegGS achieves comparable reconstruction quality to MASt3R with significantly lower memory consumption. Further optimization of MW_2 computation remains a direction for future work.

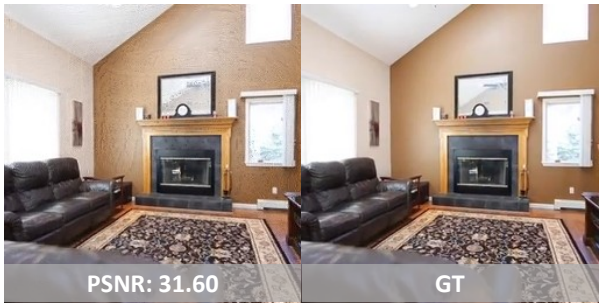

Figure 10. Visual Comparison. It is noticeable that there are many prominent noise points on the wall. Our method, in certain scenes, may produce high PSNR values, but visually, there are clearly visible noise artifacts.

9. Additional Limitations

Figure 9 demonstrates that NoPoSplat generates suboptimal Gaussians in certain scenes. In the depicted scenario, NoPoSplat struggles to accurately estimate the depth information of the reflective surface, causing the gaussians to fail at capturing the spatial geometry effectively. RegGS relies on the quality of the Gaussian model generated by the upstream model, and abnormal gaussians introduced during scene fusion can lead to errors.

Figure 10 shows that, in certain scenes, while our method achieves high PSNR values, there are noticeable noise artifacts. These noise points are likely introduced during the refinement stage or could be a result of the low image resolution used in our quantitative evaluation. In future work, we will attempt to address this issue.