Supplementary material for Example-Based Sampling with Diffusion Models

Bastien Doignies
Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS
France
bastien.doignies@liris.cnrs.fr

Nicolas Bonneel
Univ Lyon, CNRS, UCBL, INSA Lyon, LIRIS
France
nicolas.bonneel@liris.cnrs.fr

David Coeurjolly
Univ Lyon, CNRS, UCBL, INSA Lyon, LIRIS
France
david.coeurjolly@liris.cnrs.fr

Julie Digne
Univ Lyon, CNRS, UCBL, INSA Lyon, LIRIS
France
julie.digne@liris.cnrs.fr

Lois Paulin
Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, Adobe
France
paulin@adobe.com

Jean-Claude Iehl
Univ Lyon, CNRS, UCBL, INSA Lyon, LIRIS
France
jean-claude.iehl@liris.cnrs.fr

Victor Ostromoukhov
Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS
France
victor.ostromoukhov@liris.cnrs.fr

ACM Reference Format:
Bastien Doignies, Nicolas Bonneel, David Coeurjolly, Julie Digne, Lois Paulin, Jean-Claude Iehl, and Victor Ostromoukhov. 2023. Supplementary material for Example-Based Sampling with Diffusion Models. In *Proceedings of SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23)*. ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Contents

1 DIFFUSION MODEL

Diffusion models date back to the work of Sohl-Dickstein et al. [2015] but were popularized by Ho et al. [2020] for image synthesis. This section recalls the details for completeness.

Probabilistic Denoising Diffusion models involve a *forward* process, where noise is gradually added to the signal (here an image) and a *reverse* process where noise is removed through a learnable network. The forward diffusion process is a Markov Chain, where each transition adds Gaussian Noise to the image, following:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)\,, \tag{1}$$

where $(\beta_t)_{t=0}^T$ are the noise variances for each time t. The variance schedule is chosen such that nothing distinguishes x_T from a white noise. In our model, we set the variances β_t to follow a linear schedule.

One has:

$$q_{x_{1:T}|x_0} = \prod_{t=1\cdots T} q(x_t|x_{t-1})\,. \tag{2}$$

The reverse (denoising) process is also a Markov Chain, with transitions:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))\,, \tag{3}$$

μ_θ and Σ_θ are learned by examples. To simplify, following the work of Ho et al. [2020], we consider that $\Sigma_\theta = \sigma_t I$, with $\sigma_t = \beta_t$. The forward process allows to sample x_t with arbitrary t from x_0, following:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar\alpha_t}x_0, (1-\bar\alpha_t)I)\,, \tag{4}$$

with $\alpha_t = 1 - \beta_t$ and $\bar\alpha_t = \prod_{s=1}^t \alpha_s$.

During training, and image x_0 is drawn from the set of examples, along with a random time $t \in 1\cdots T$, a random noise image ε is drawn following $\mathcal{N}(0, I)$ and the algorithm tries to minimize:

$$\|\varepsilon - \varepsilon_\theta(\sqrt{\bar\alpha_t}x_0 + \sqrt{1-\alpha_t}\varepsilon, t)\|^2\,, \tag{5}$$

by gradient descent.

During sampling a random noise image $z \sim \mathcal{N}(0, I)$ is drawn and iteratively denoised by applying:

$$x_{t-1} = \frac{1}{\sqrt{\bar\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\varepsilon_\theta(x_t, t)) + \sigma_t z\,, \tag{6}$$

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

where z is a random noise and in our case, we take $\sigma_t = \beta_t$ The key ingredient of diffusion models is the approximator ε_θ, which is modeled by a neural network.

2 NETWORK

Our network is a slightly modified version of the denoising network of Ho et al. [2020] and is summarized on Figure 1.As for hyper-parameters, we used a fixed linear variance schedule from 10^{-4} to 10^{-2} with 1000 diffusion steps at training, similarly to Ho et al. [2020].

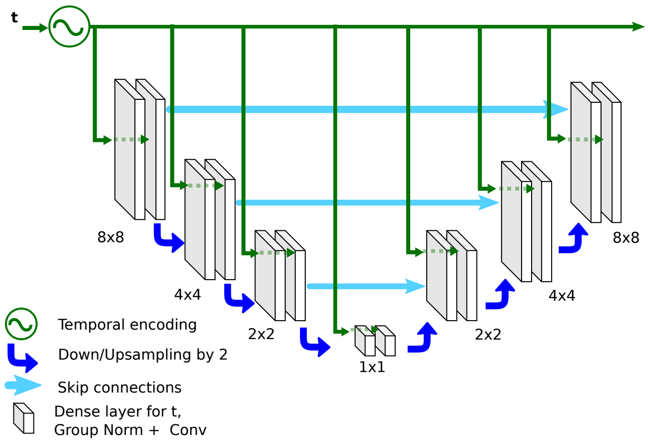


Figure 1: Denoising network architecture

Class conditioned variant. We also introduce a modified network able to learn multiple classes by conditioning it with the name of the distribution class. To do so, we encode the class through one hot encoding followed by an MLP, whose parameters are learned together with the denoising network parameters. All pixels in the input noise image are then concatenated with the resulting vector. Note that the condition vector is constant over the whole image since it encodes the class of the desired sample distribution.

3 ADDITIONAL UPSCALING COMPARISONS TO [Huang et al. 2022] AND [Tu et al. 2019]

In Figure 2, we provide additional upscaling results to 4096 samples using Huang et al. [2022] and Tu et al. [2019] when targeting an Sobol'+Owen point pattern. Both Huang et al. [2022] and Tu et al. [2019] are not able to upsample the point set while preserving the statistical properties. On the contrary, our approach provides a much more reliable upsampling.

4 COMPLETE EVALUATION OF THE CLASS CONDITIONED VARIANT

In Figures 3, 4, 5, 6, 7, we compare, for all metrics, the class conditioned variant (*i.e.* trained on all classes with conditioning) with the results obtained by single trained class models. While the class conditioned model performs slightly worse than our per class vanilla architecture, the various metrics show that it is still a good approximation of the various samplers. The most difficult property to capture seems to be the generalized L2 discrepancy of the distribution, but even so, the conditioned model is not too far from the per class model.

5 ADDITIONAL RESULTS IN 3D

In Figures 8, 9, 10, 11 and 12, we provide additional preliminary results for 3d point set synthesis using our network. Our model successfully captures key properties of the learned point sets.

6 ABLATION STUDY: NUMBER OF DIFFUSION STEPS

In Figures 3, 4, 5, 6, 7, we compare, for all metrics, the quality of our model when changing the number of diffusion steps in {50, 100, 250, 500, 1000} when generating the point sets (the learning stage is still performed with 1000 diffusion steps). The inference computational cost is linear with the number of steps, and we have observed that using only 50 steps is a good trade-off between quality and efficiency.

7 ABLATION STUDY: EFFECT OF THE DATABASE SIZE

As discussed in the main paper, the number of examplars used for the training may have an impact on the model quality. In figures 18, 19, 20, 21 and 22, we compare for all metrics the quality of the proposed model on a training set of 64k realizations and a training set of 32 realizations. The model seems to perform well as reported by the computed metrics, but further experiments would be necessary to assess the generalization power of the model trained on 32-examples datasets.

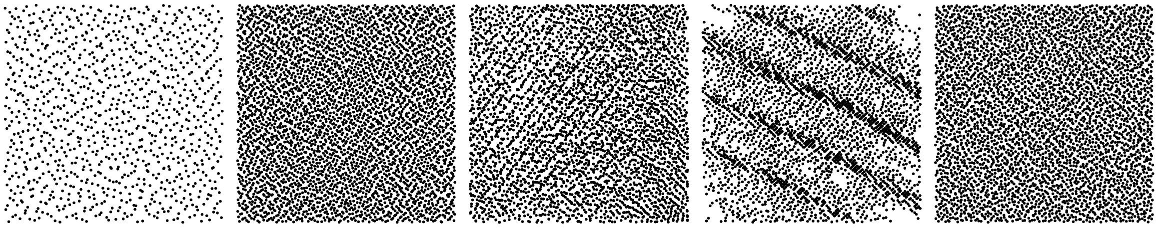
REFERENCES

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

Xingchang Huang, Pooran Memari, Hans-Peter Seidel, and Gurprit Singh. 2022. Point-Pattern Synthesis using Gabor and Random Filters. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 169–179.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2256–2265.

Peihan Tu, Dani Lischinski, and Hui Huang. 2019. Point Pattern Synthesis via Irregular Convolution. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 109–122.



(a) Sobol'+Owen (1k) (b) Sobol'+Owen (4k) (c) [Huang et al. 2022] (d) [Tu et al. 2019] (e) Ours

Figure 2: Using a 1024 Sobol'+Owen realization (a), we use Huang et al. [2022] (c), and [Tu et al. 2019] (d), to generate a 4096 point set. Finally, (e) presents our result (refer to the main paper for quantitative comparisons. Note that the 4096 Sobol'+Owen point set (b) is just given as a reference, the Owen scrambling differs from the one used for (a).

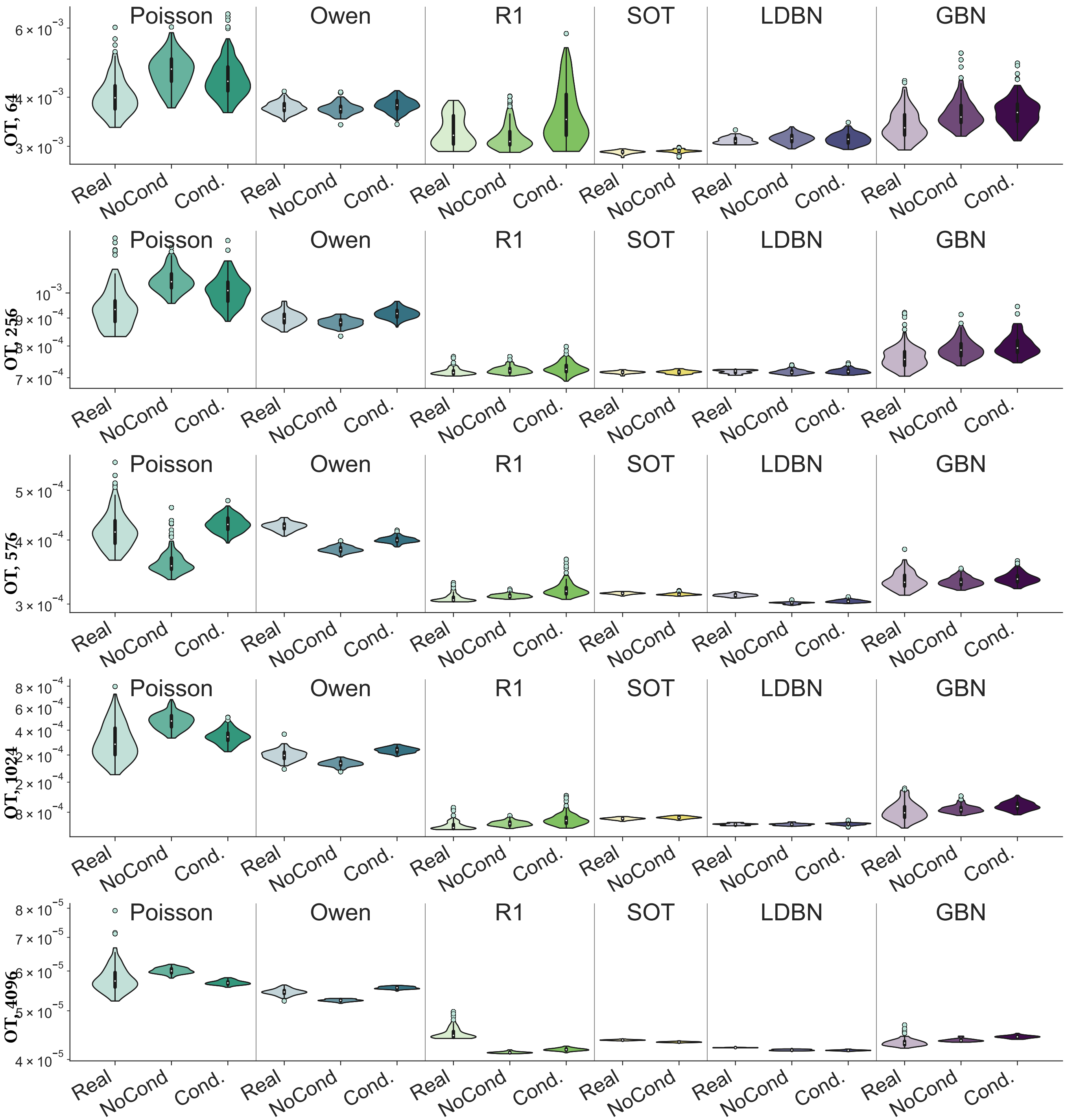


Figure 3: Evaluation of the class conditioned model - optimal transport metric.

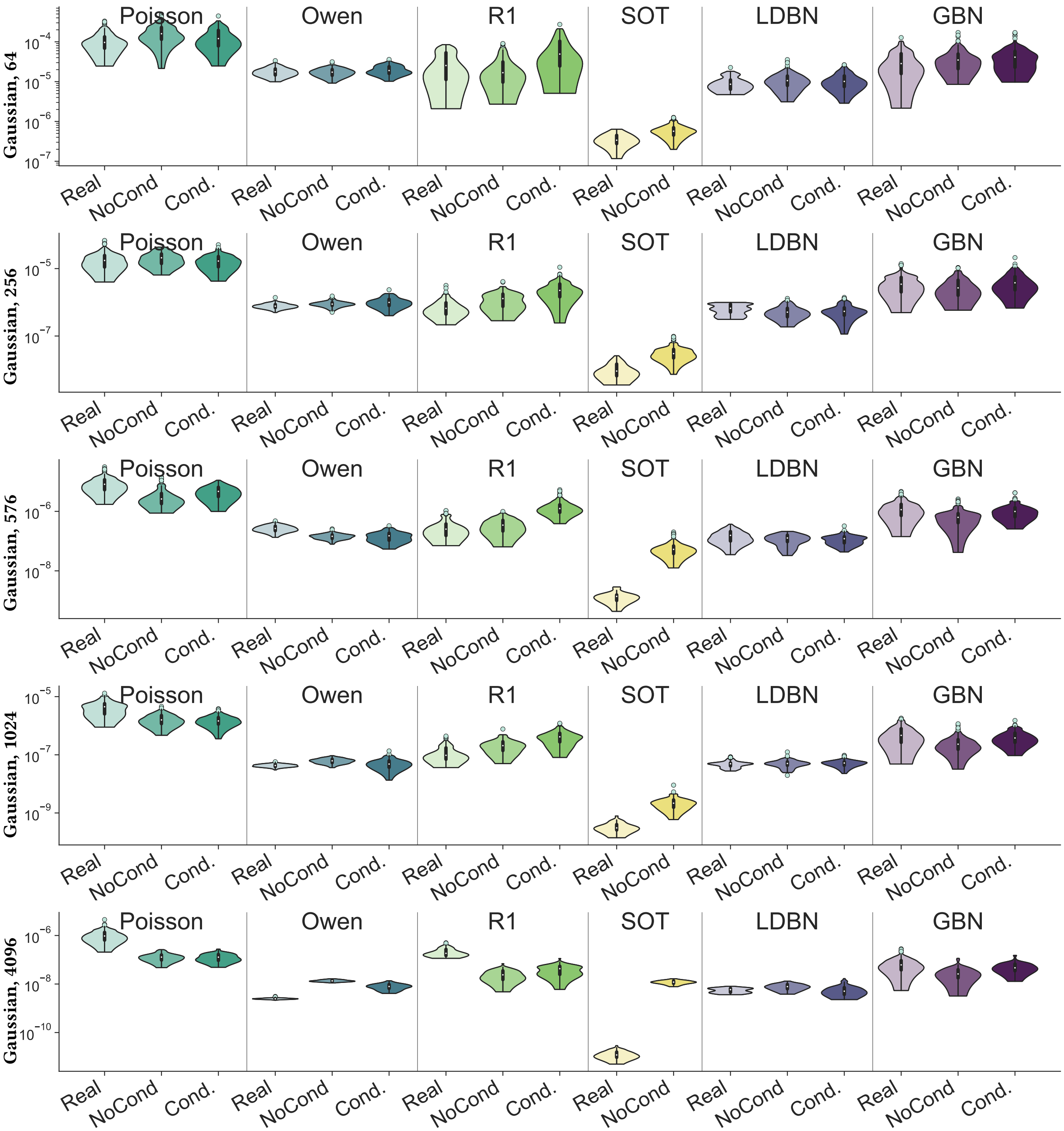


Figure 4: Evaluation of the class conditioned model - integration error on Gaussian integrands.

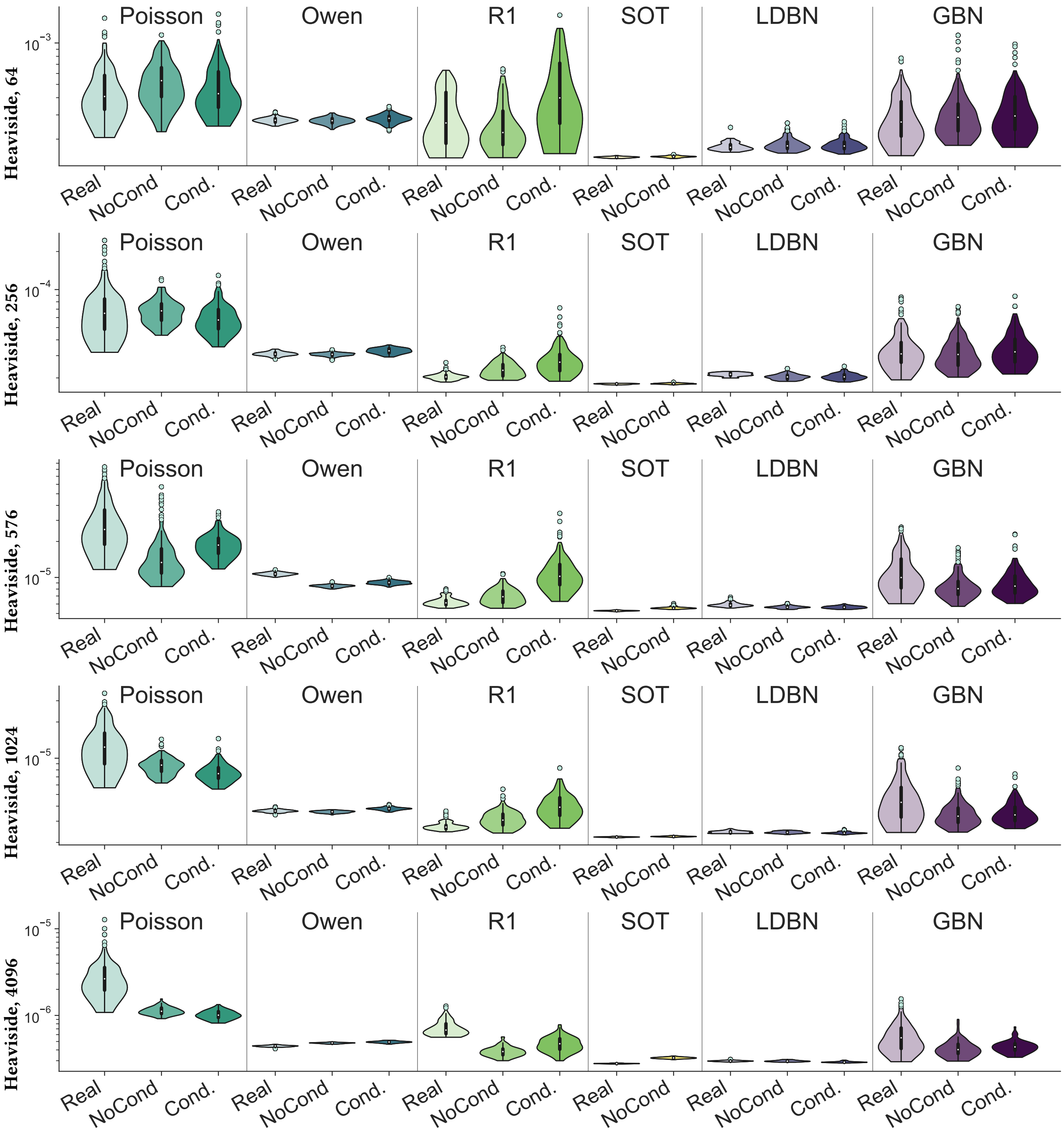


Figure 5: Evaluation of the class conditioned model - integration error on Heaviside integrands.

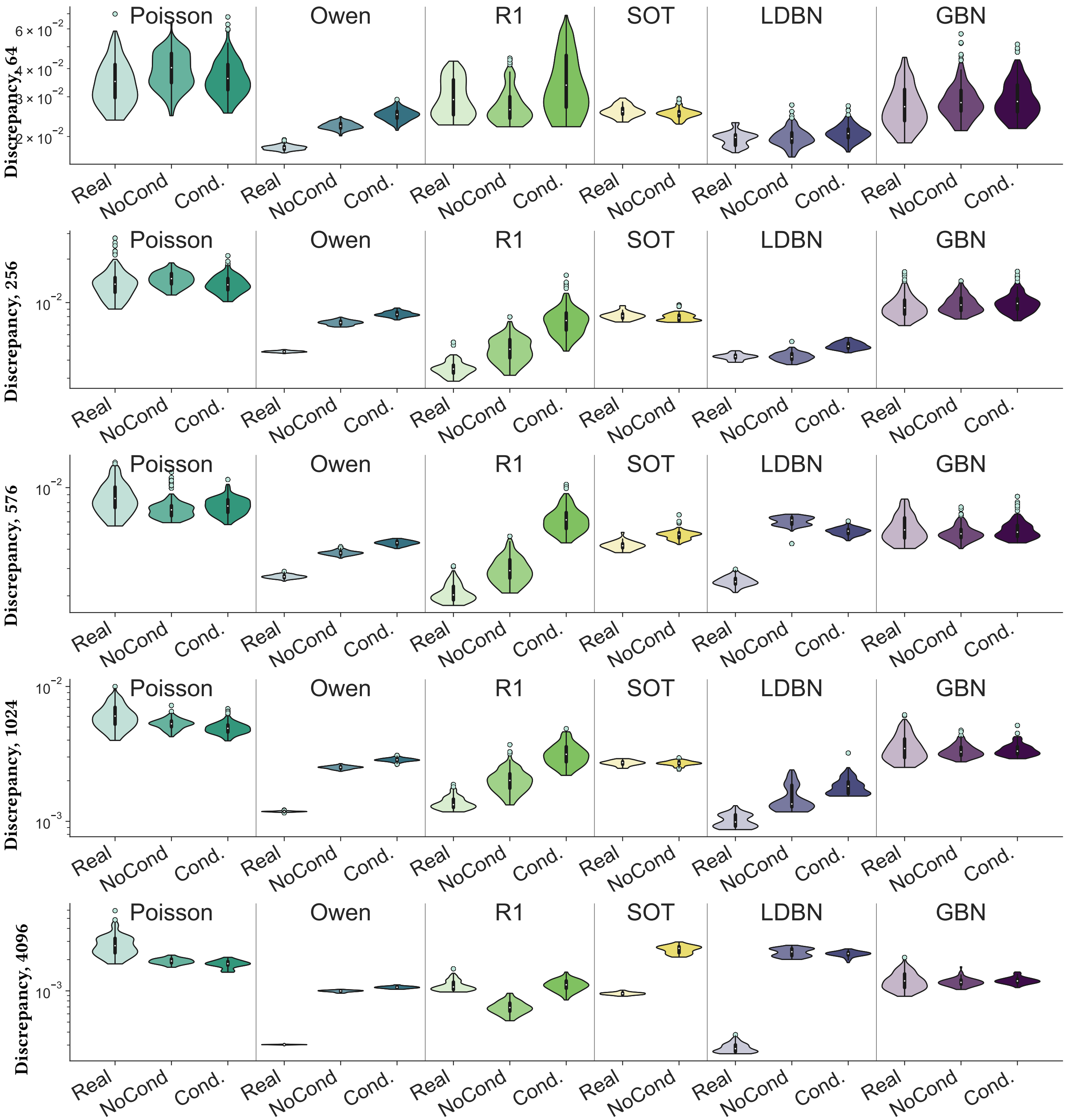


Figure 6: Evaluation of the class conditioned model - generalized L2 discrepancy.

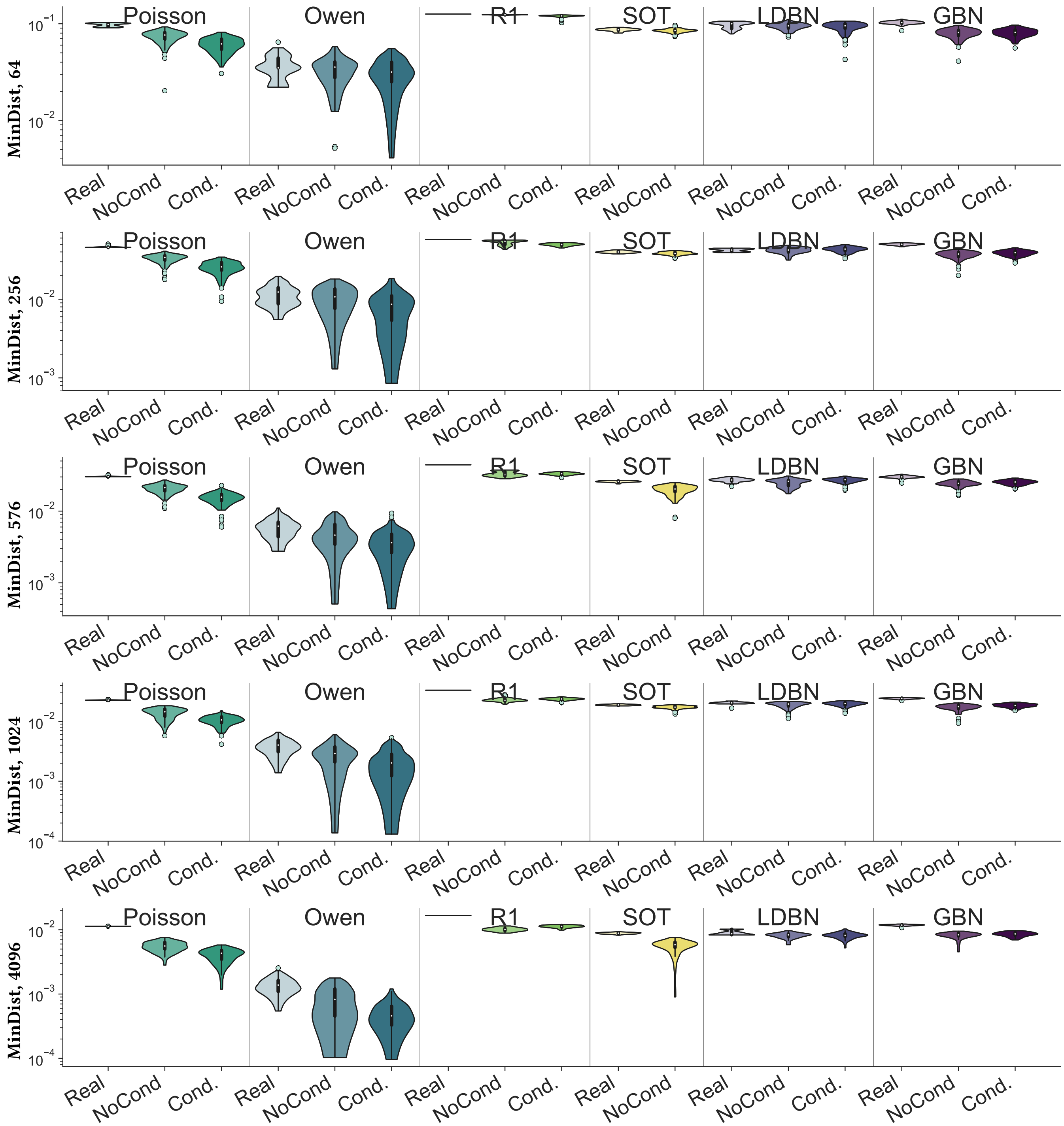


Figure 7: Evaluation of the class conditioned model - minimum pairwise distance.

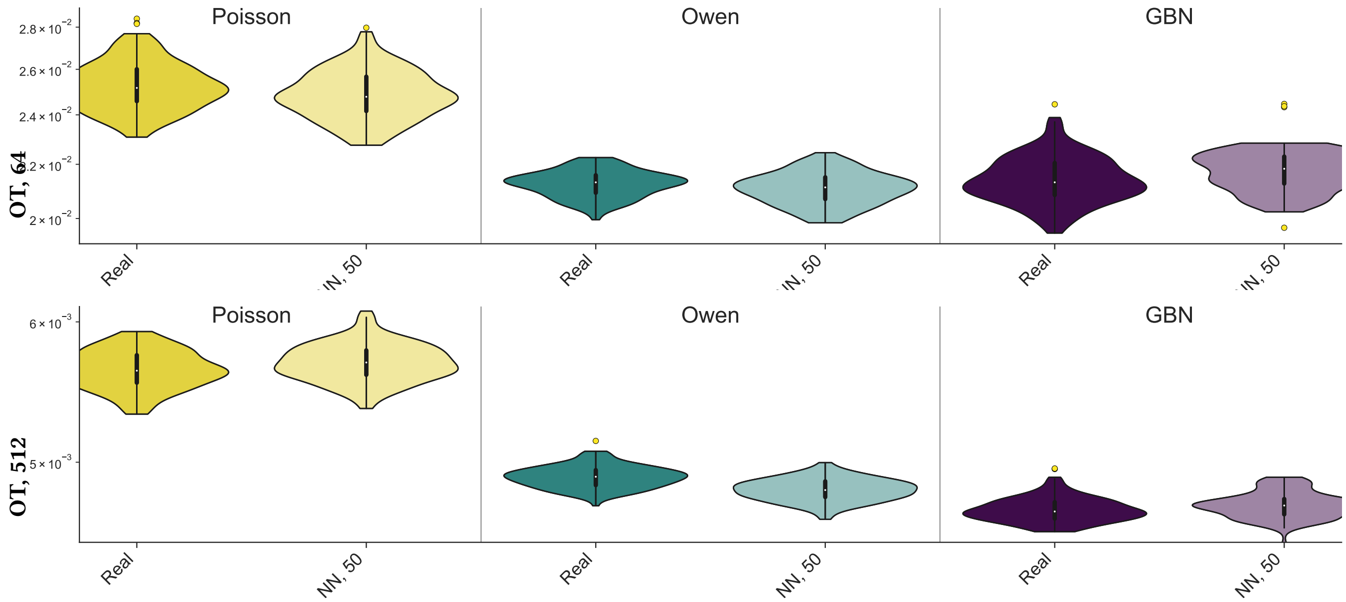


Figure 8: 3D synthesis results - optimal transport metric.

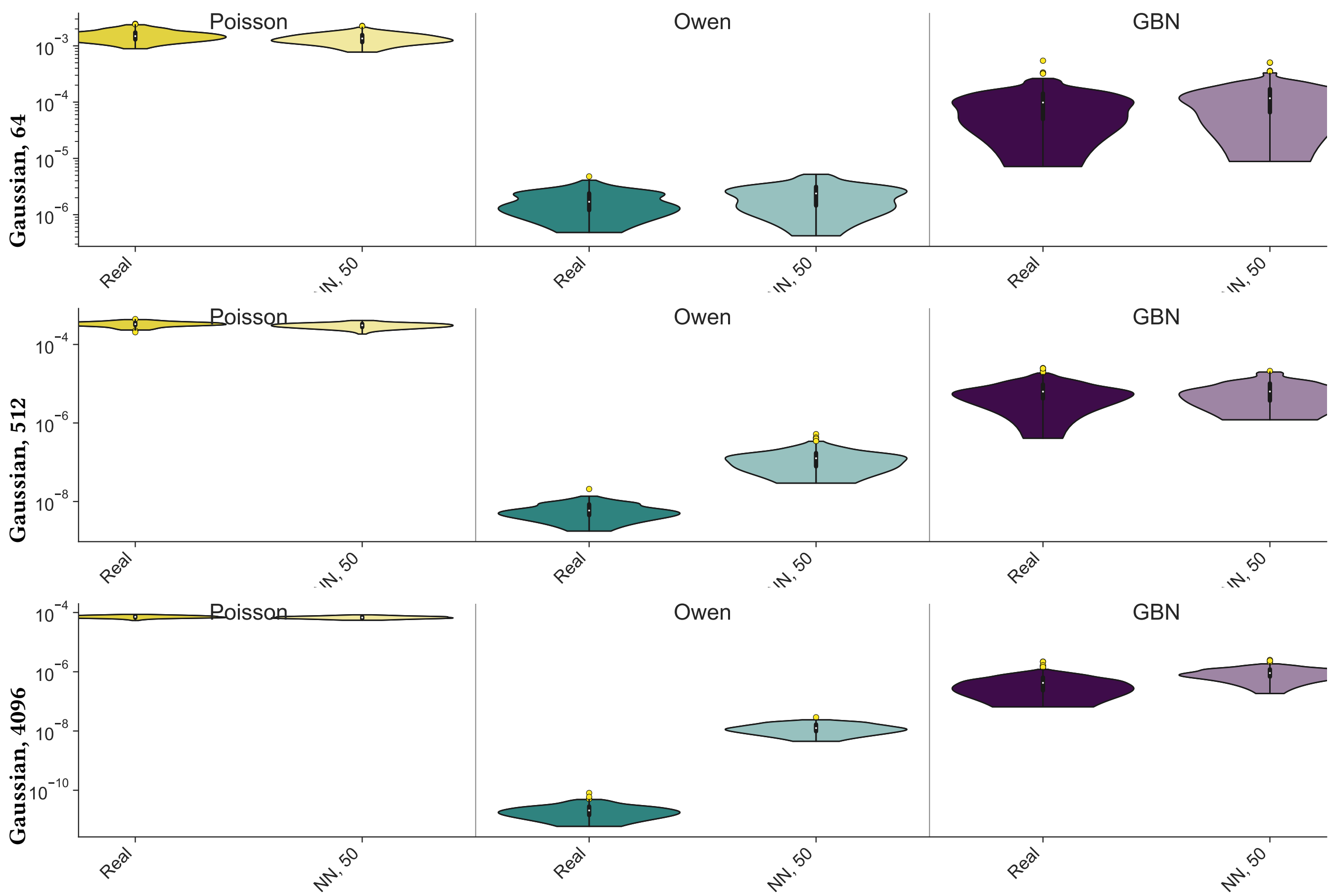


Figure 9: 3D synthesis results - integration error on Gaussian integrands.

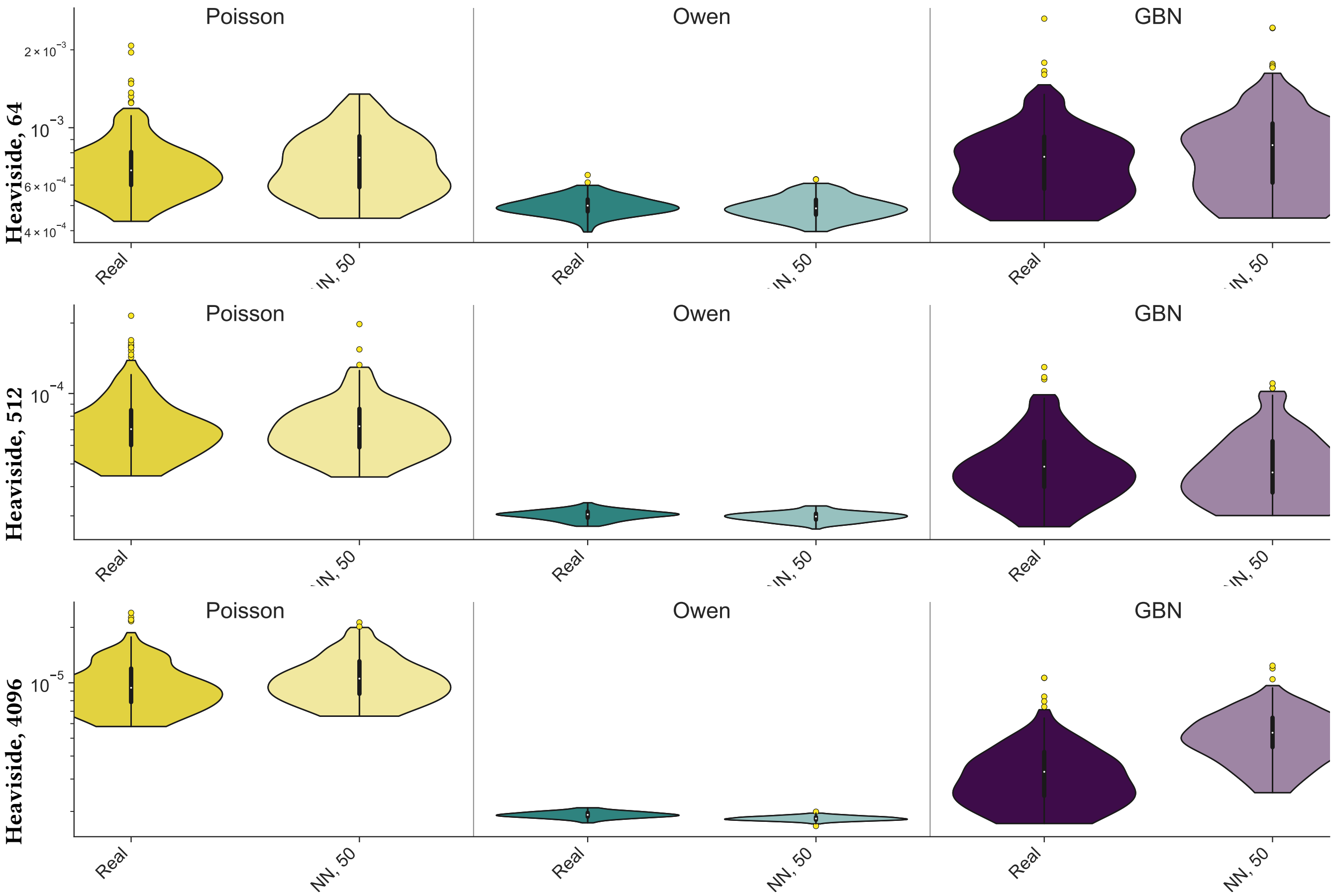


Figure 10: 3D synthesis results - integration error on Heaviside integrands.

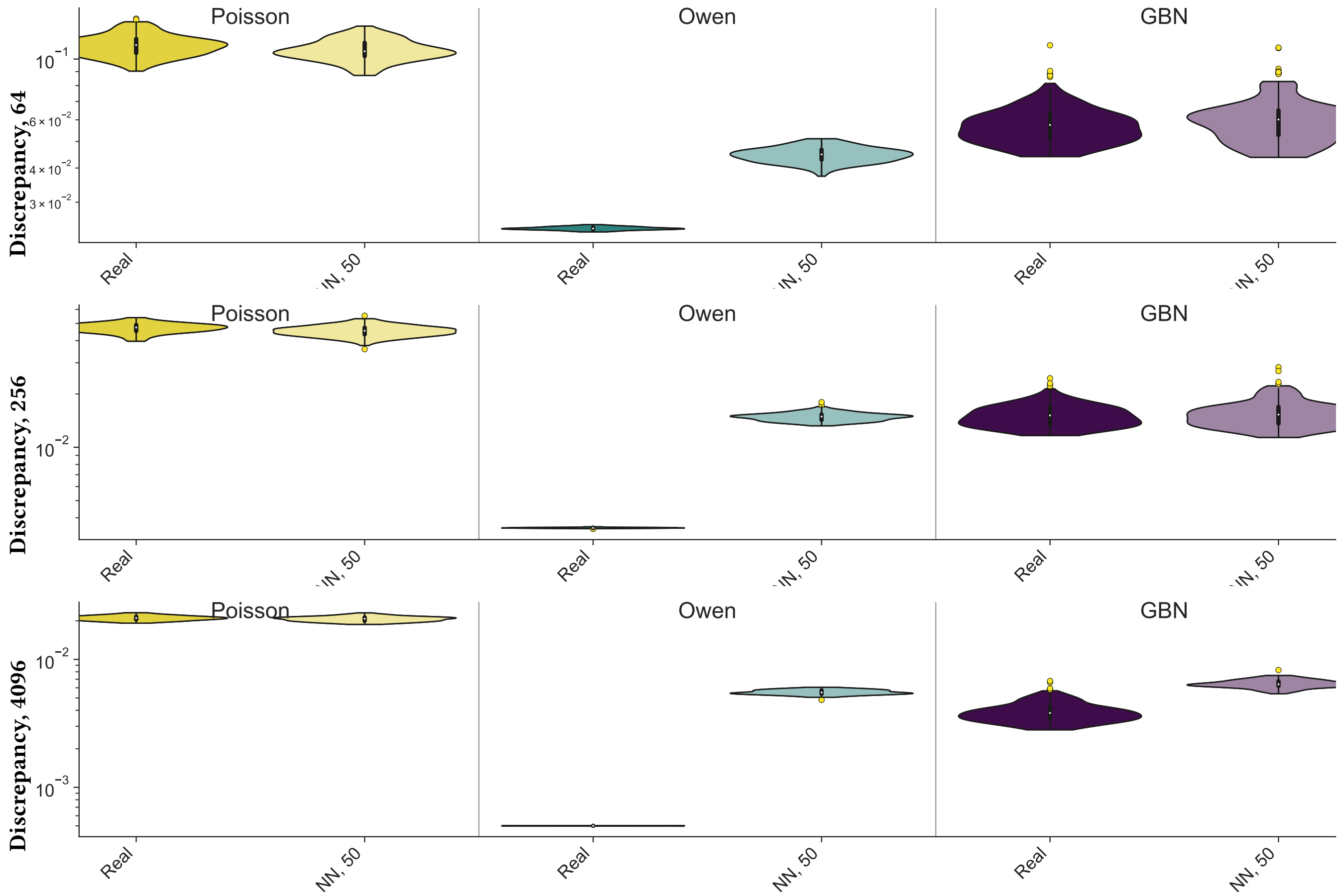


Figure 11: 3D synthesis results - generalized L2 discrepancy.

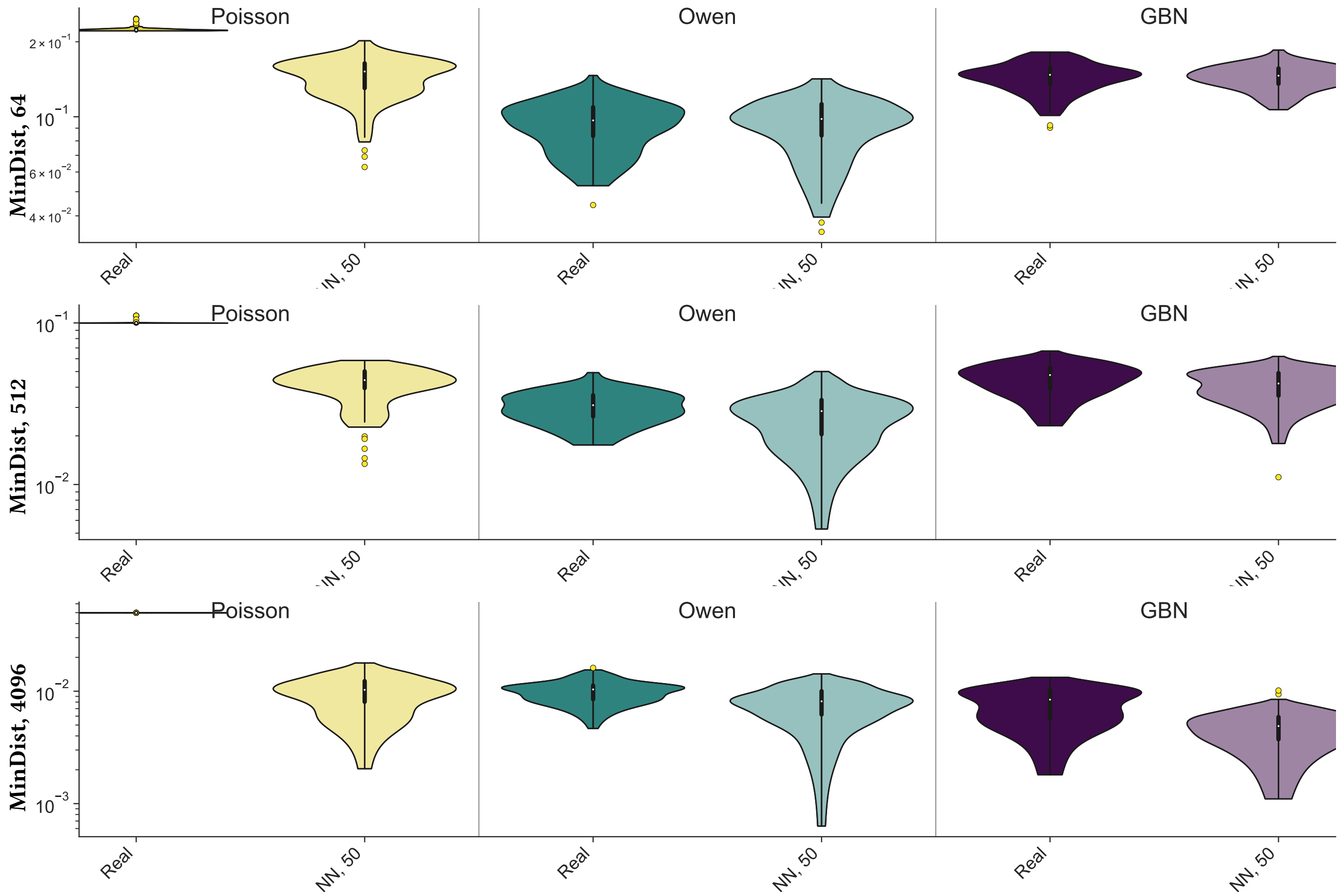


Figure 12: 3D synthesis results - minimum pairwise distance.

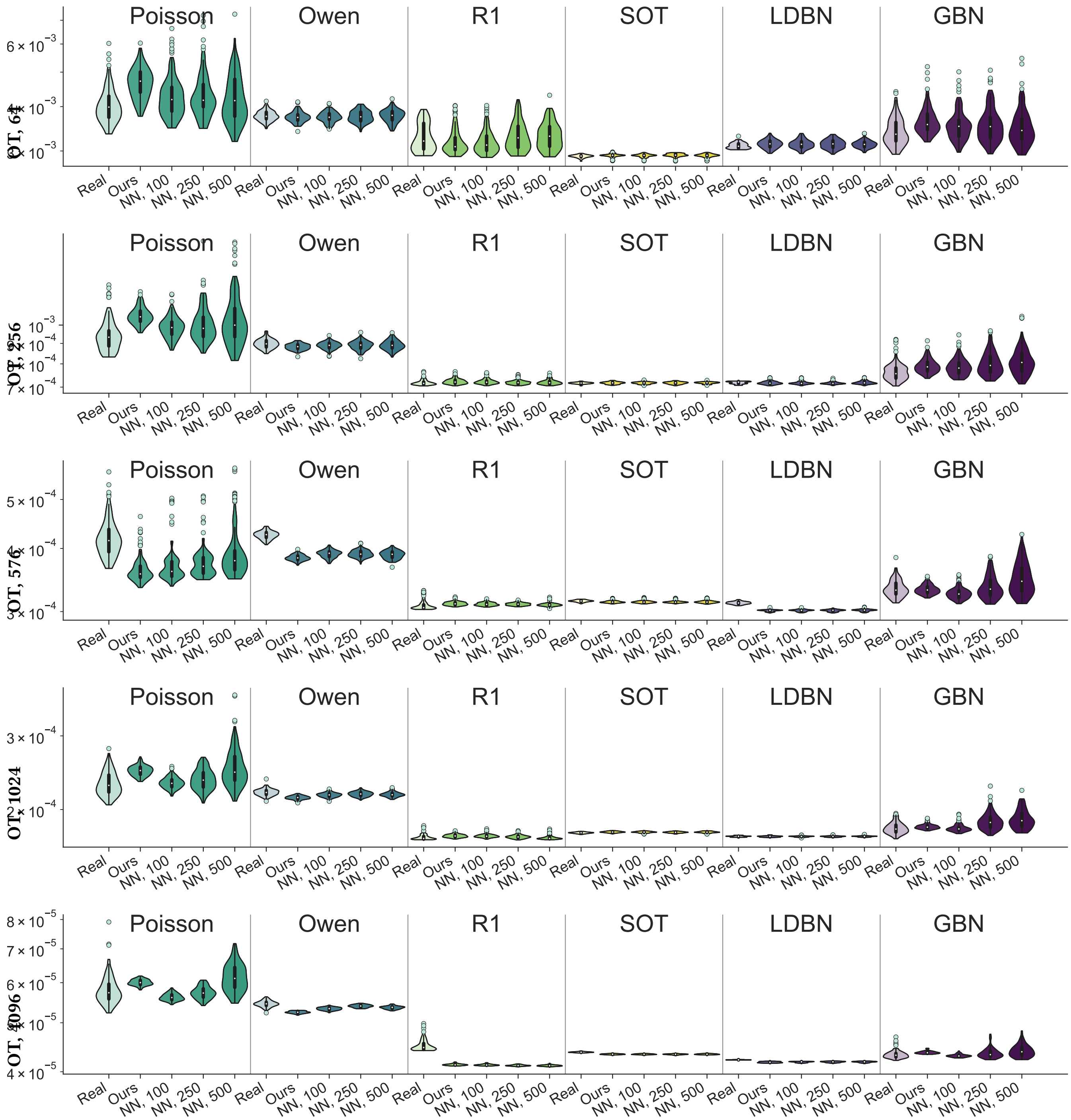


Figure 13: Effect of reducing the number of diffusion steps at inference - optimal transport metric.

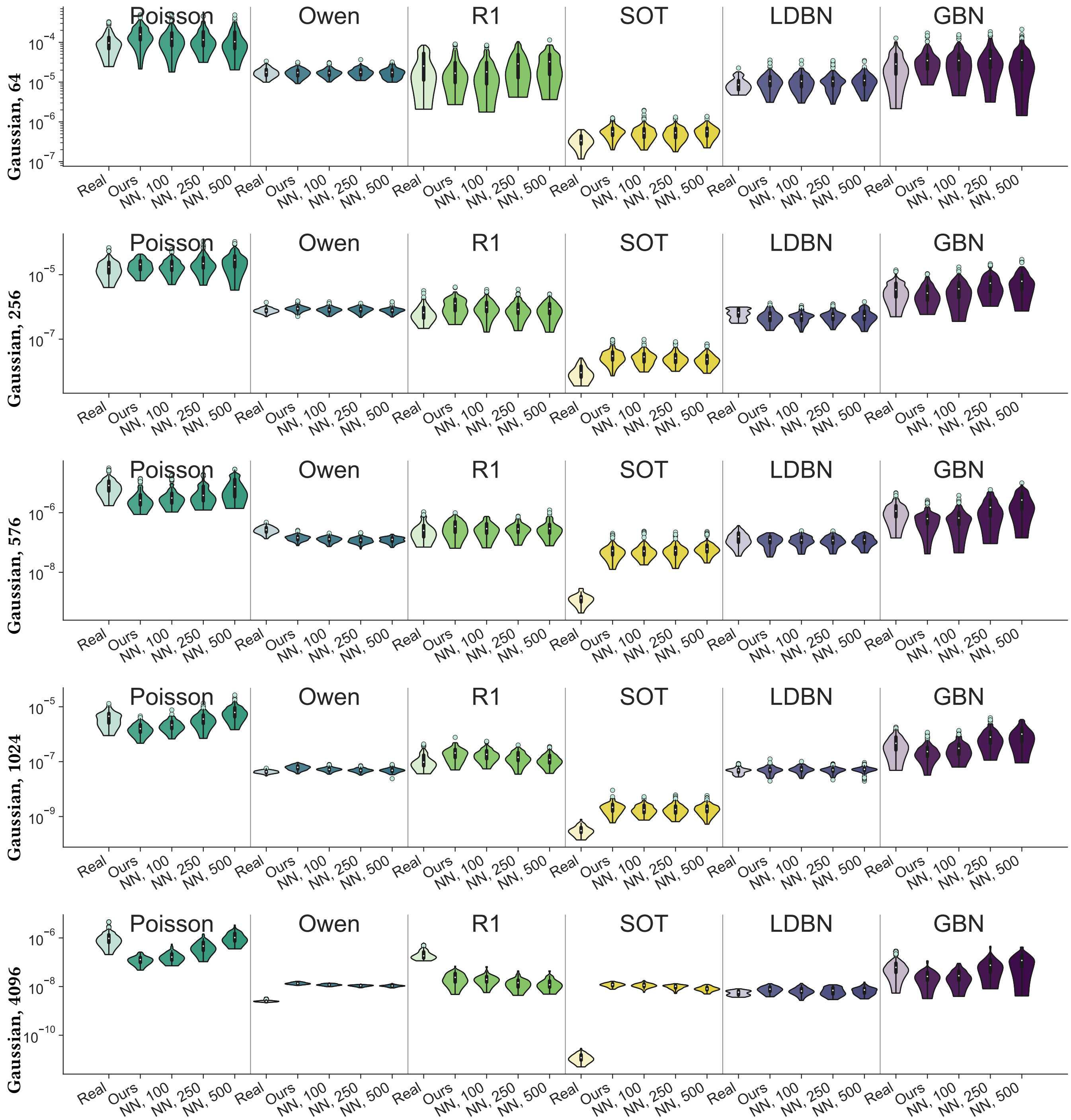


Figure 14: Effect of reducing the number of diffusion steps at inference - integration error on Gaussian integrands.

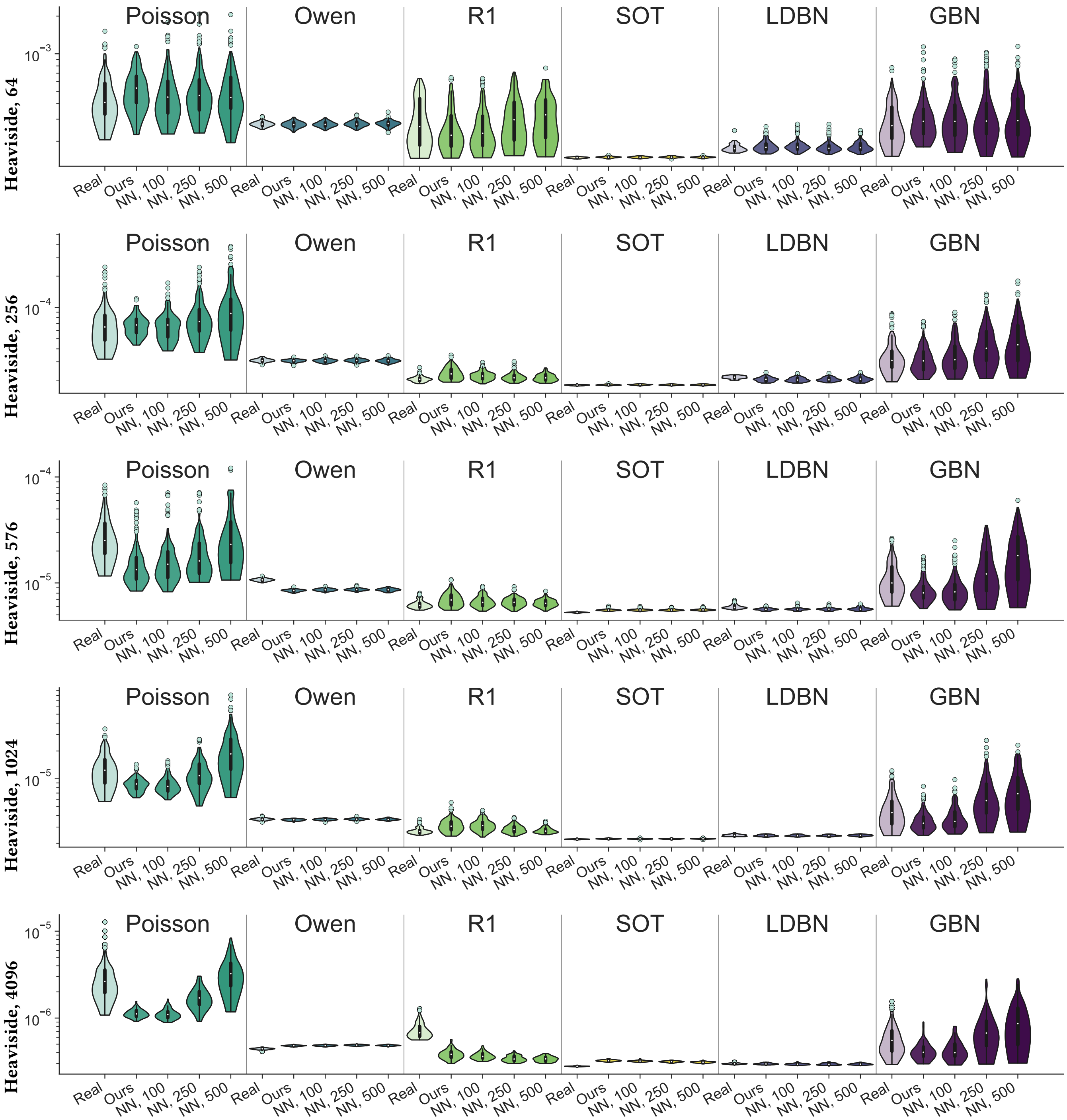


Figure 15: Effect of reducing the number of diffusion steps at inference - integration error on Heaviside integrands.

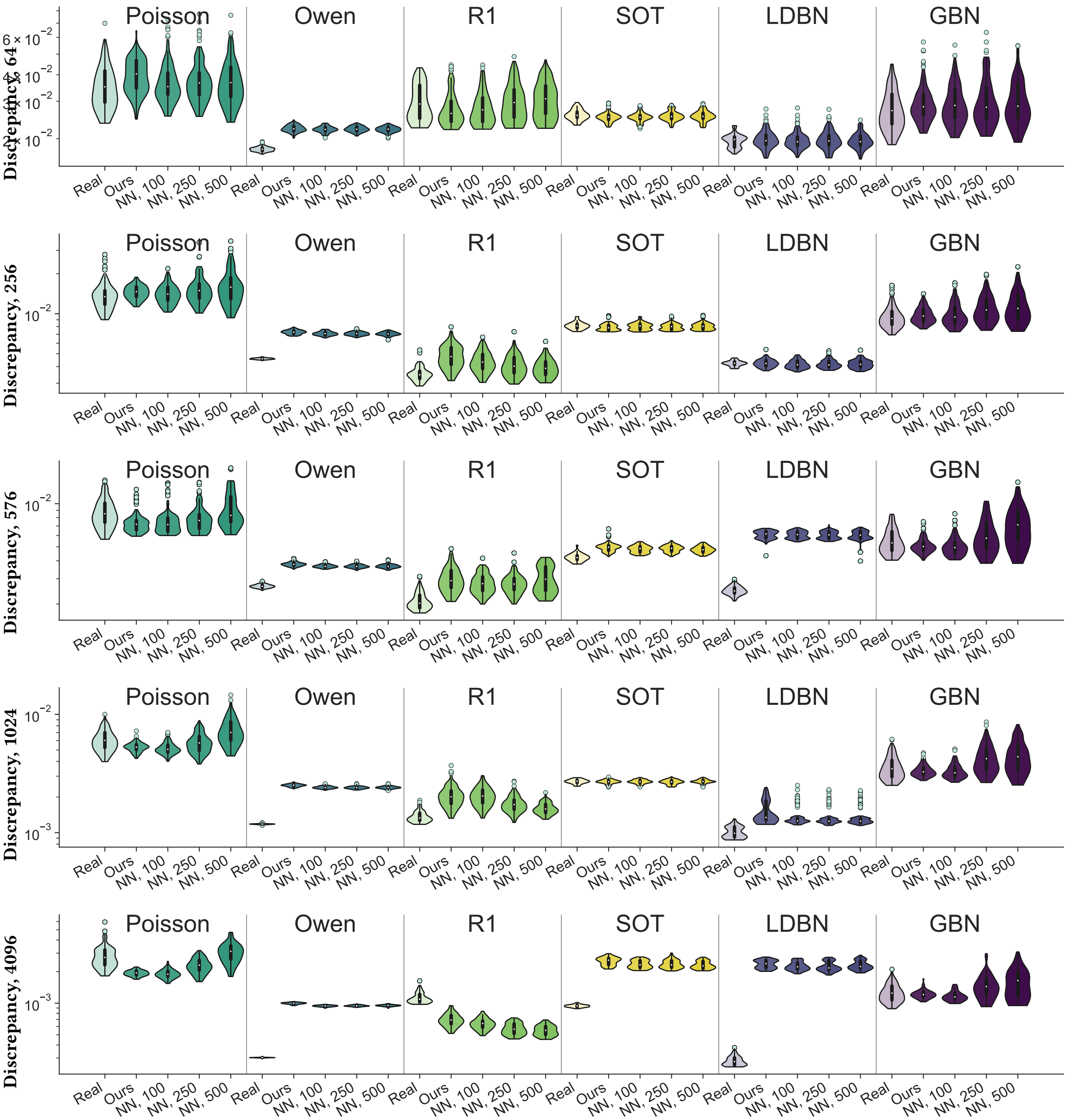


Figure 16: Effect of reducing the number of diffusion steps at inference - generalized L2 discrepancy.

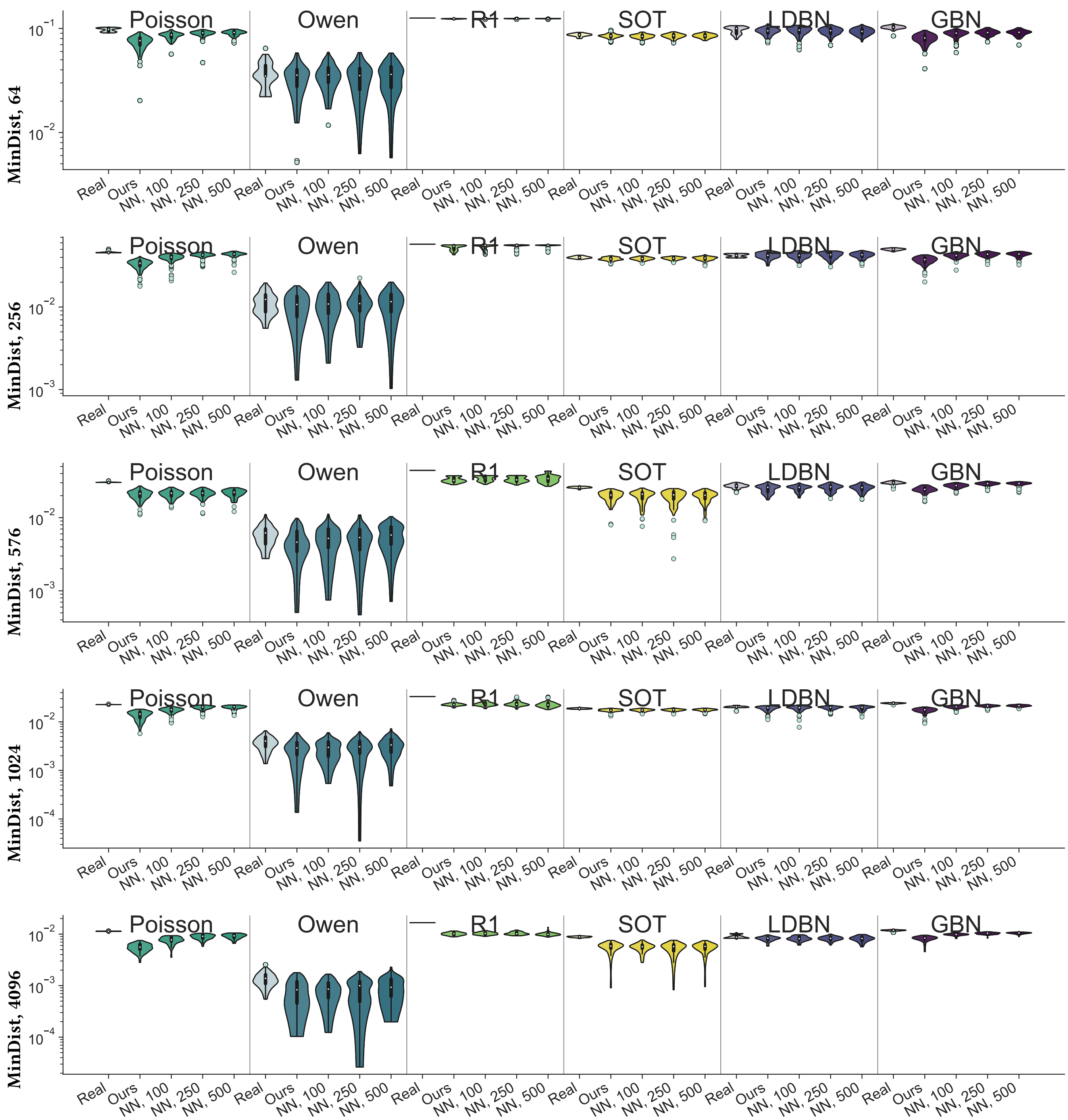


Figure 17: Effect of reducing the number of diffusion steps at inference - minimum pairwise distance test.

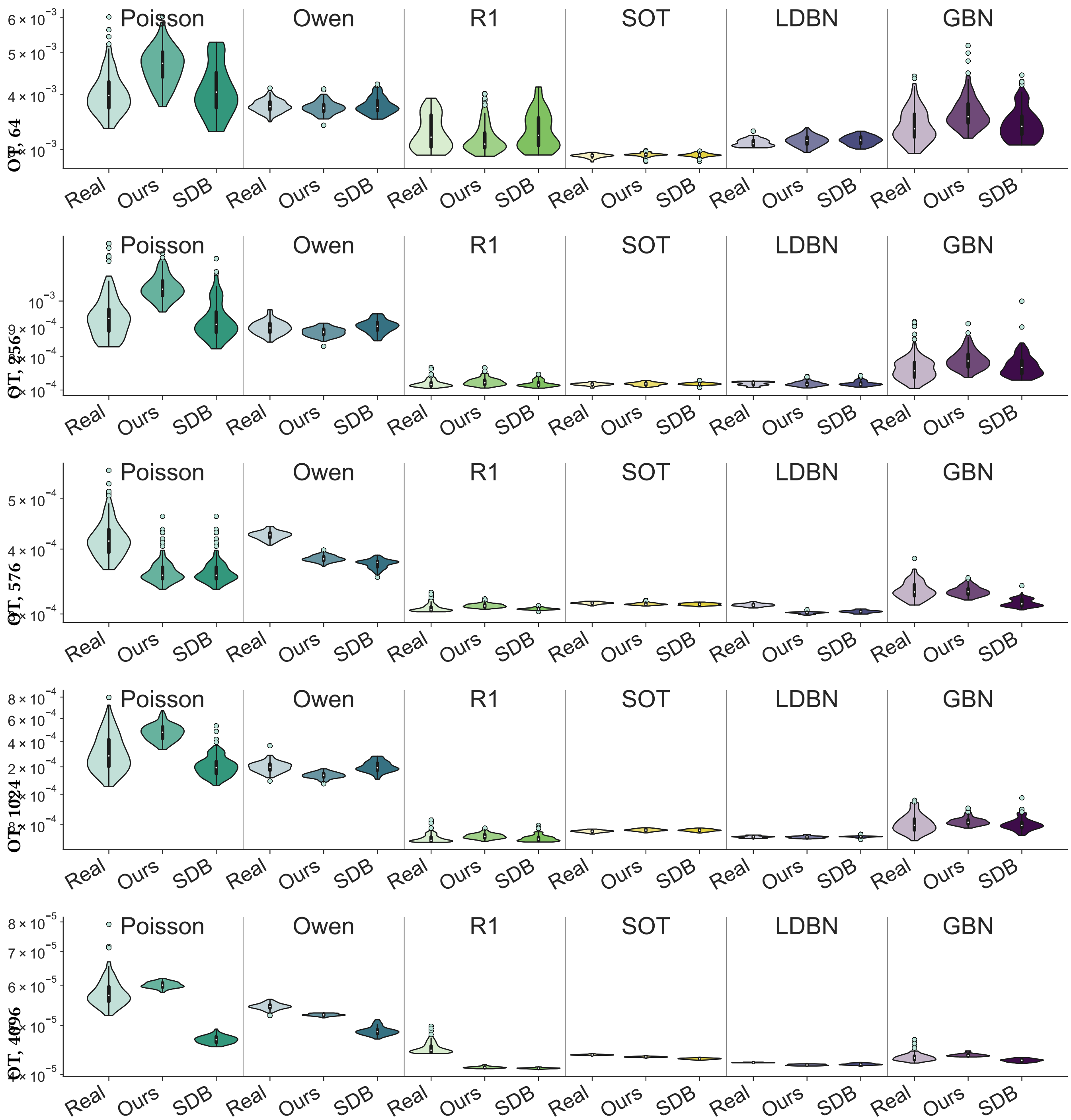


Figure 18: Effect of the database size: optimal transport metric.

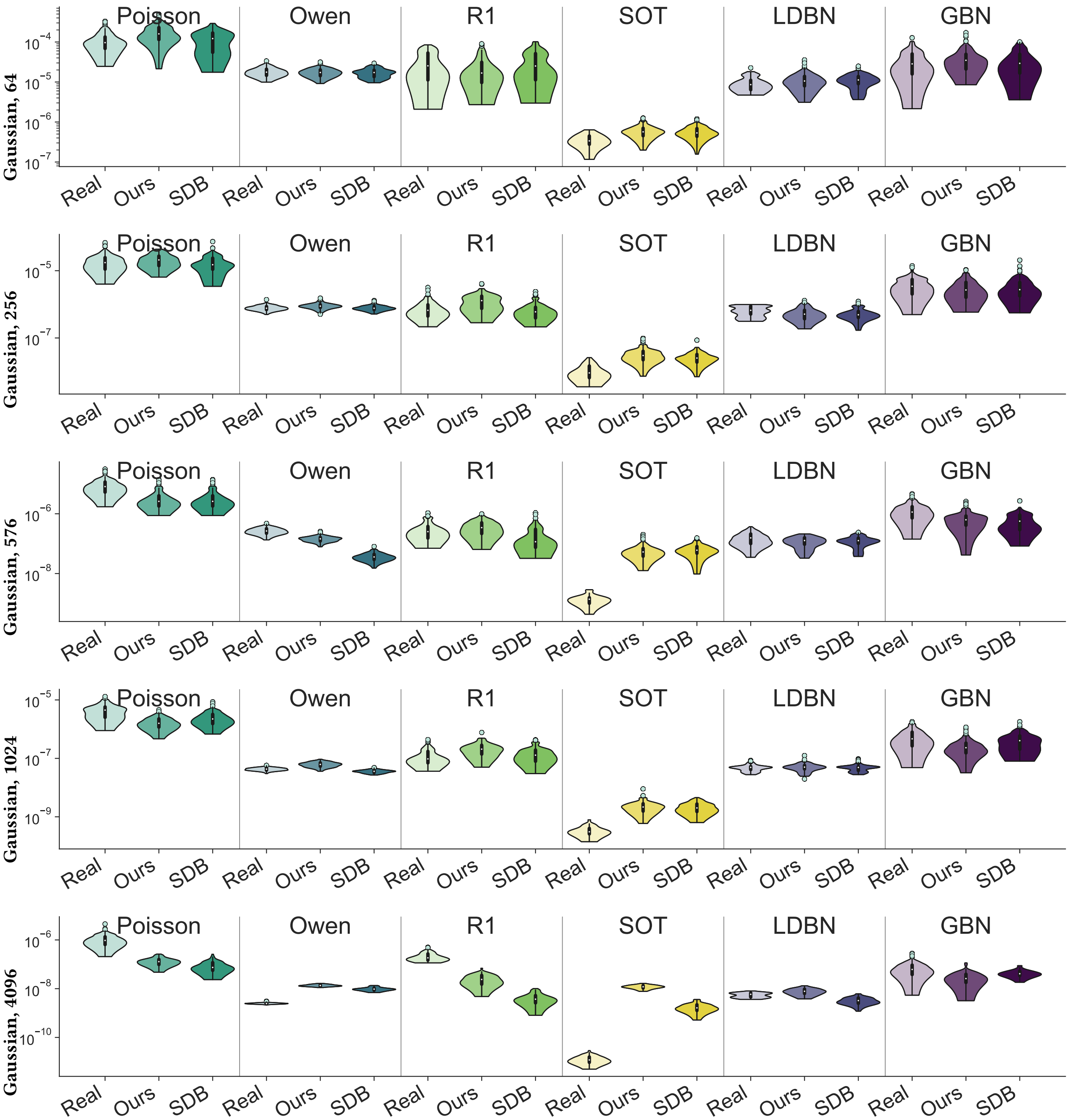


Figure 19: Effect of the database size: integration error on Gaussian integrands.

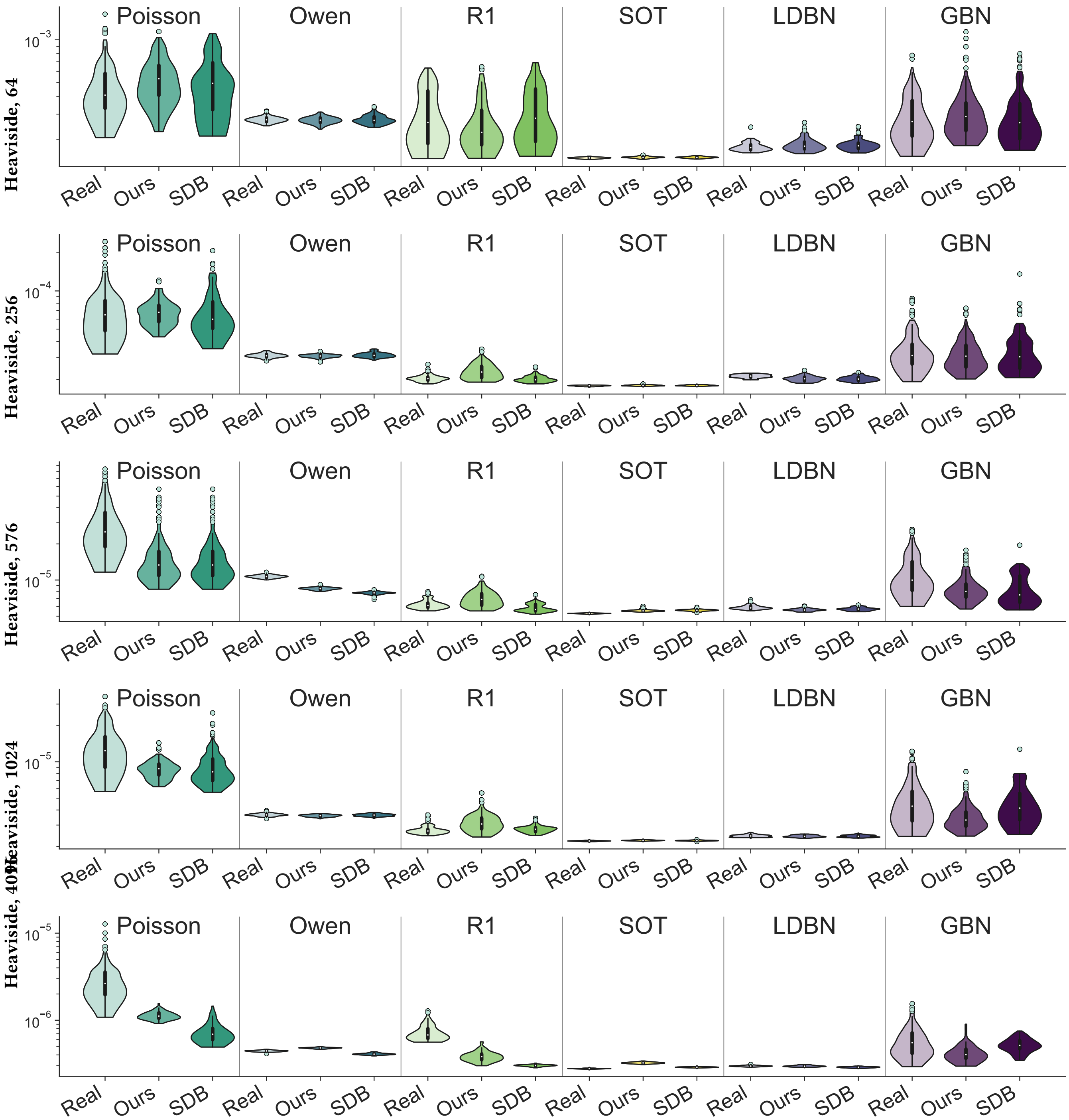


Figure 20: Effect of the database size: integration error on Heaviside integrands.

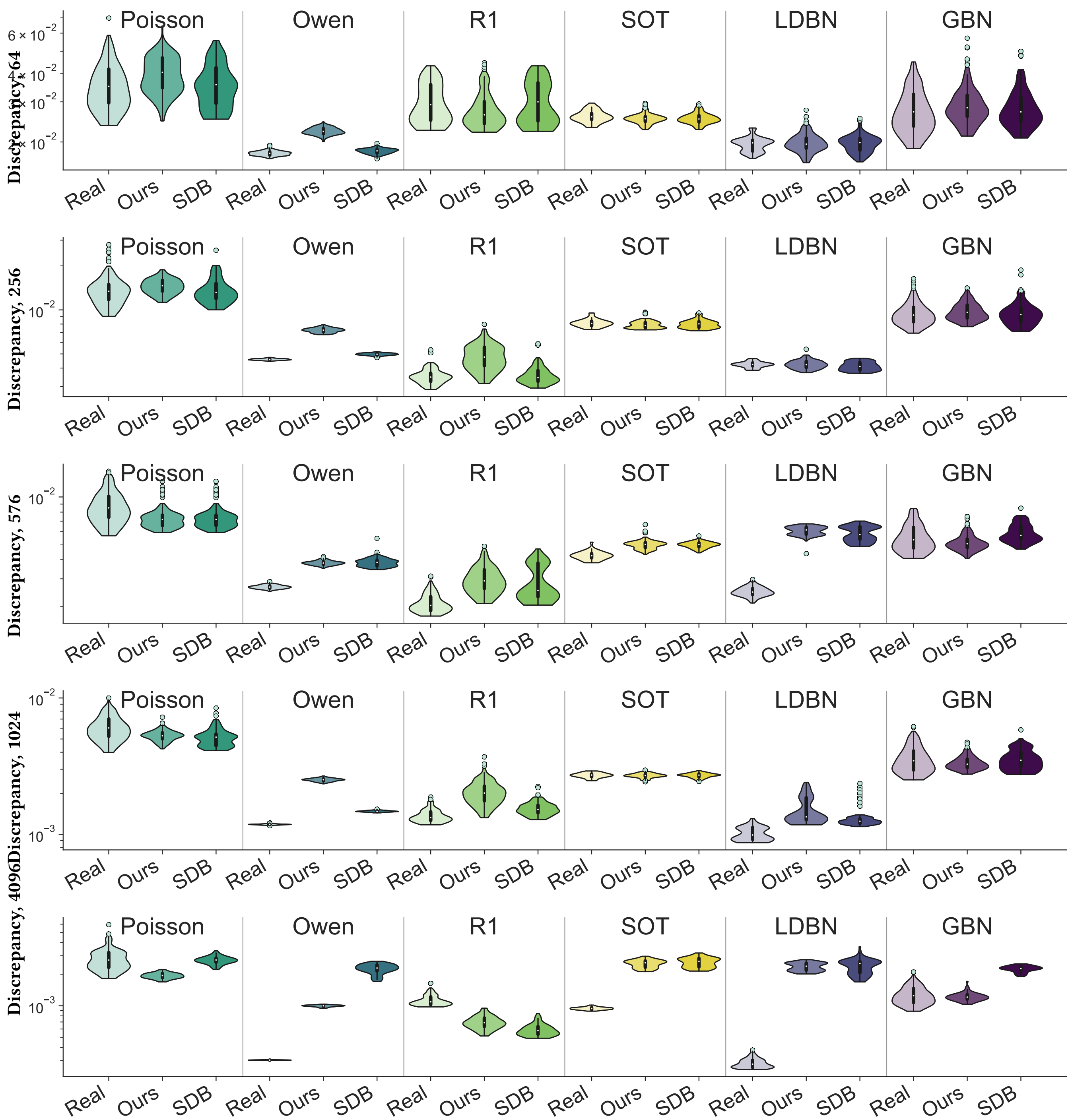


Figure 21: Effect of the database size: generalized L2 discrepancy.

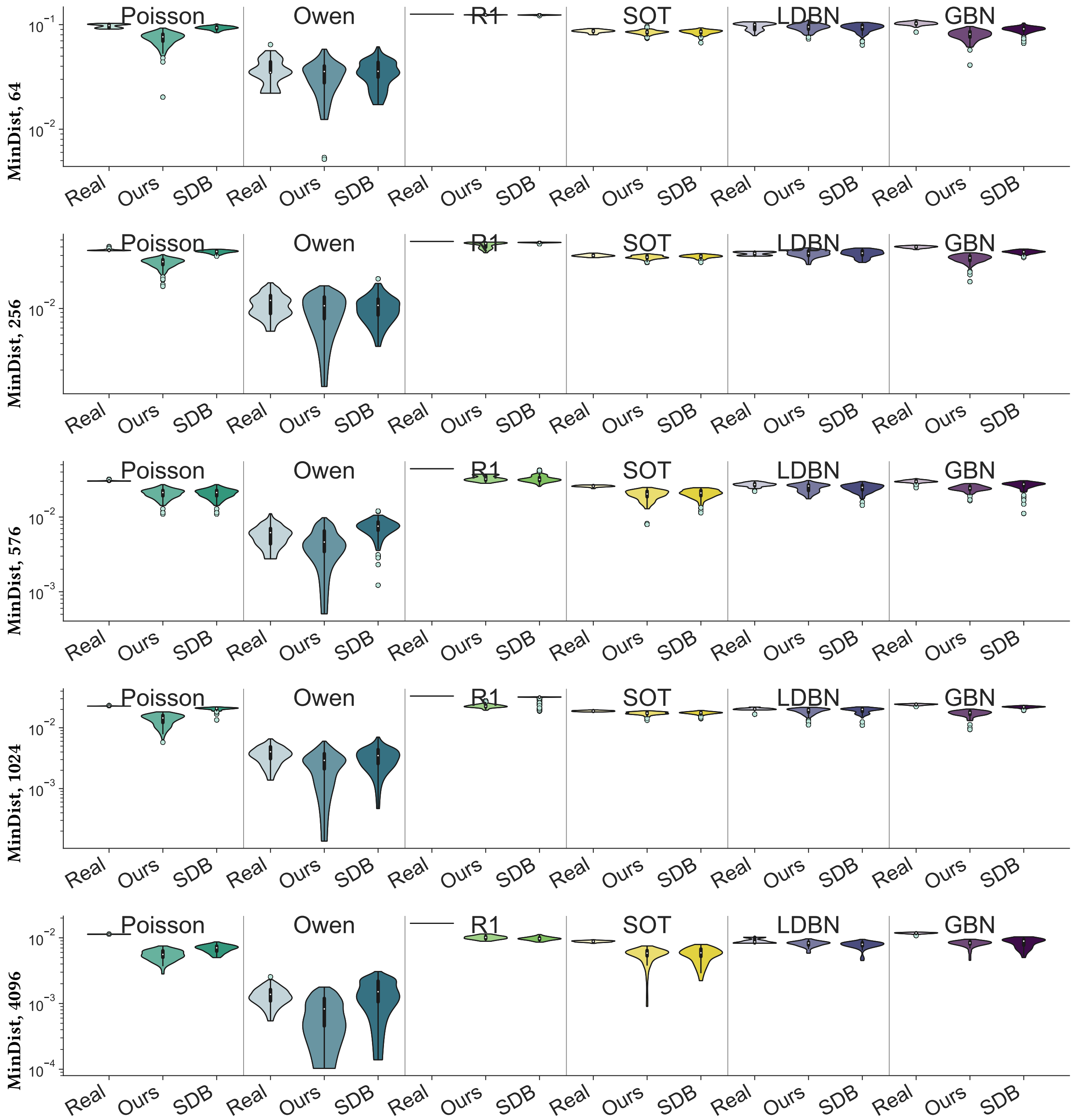


Figure 22: Effect of the database size: minimum pairwise distance test.