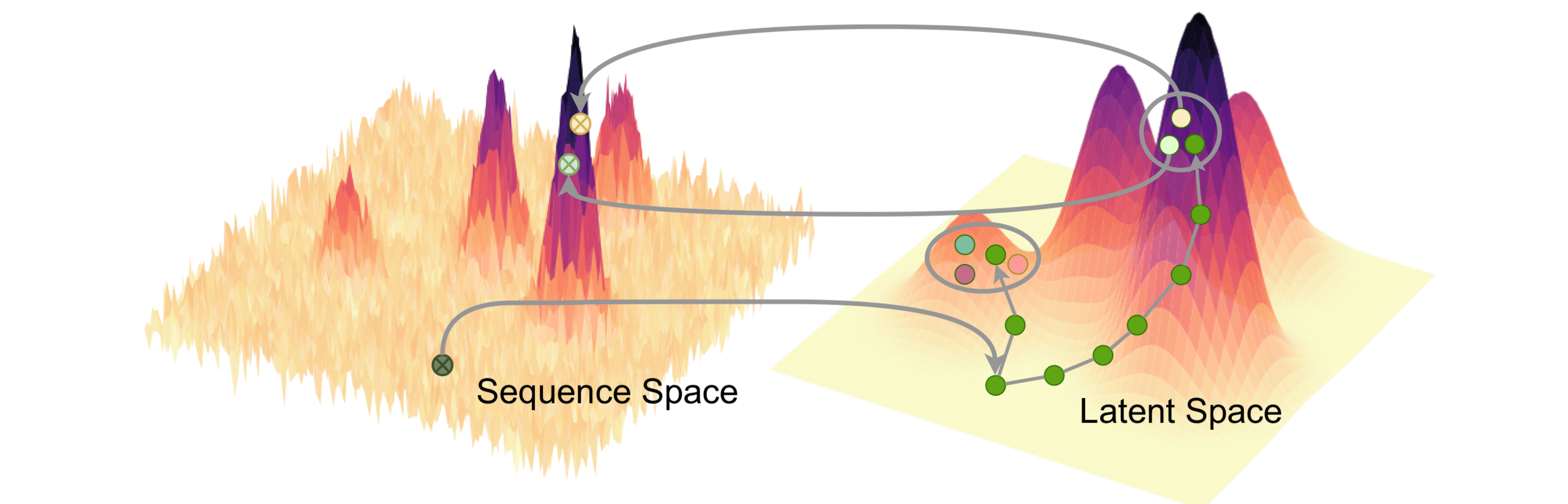
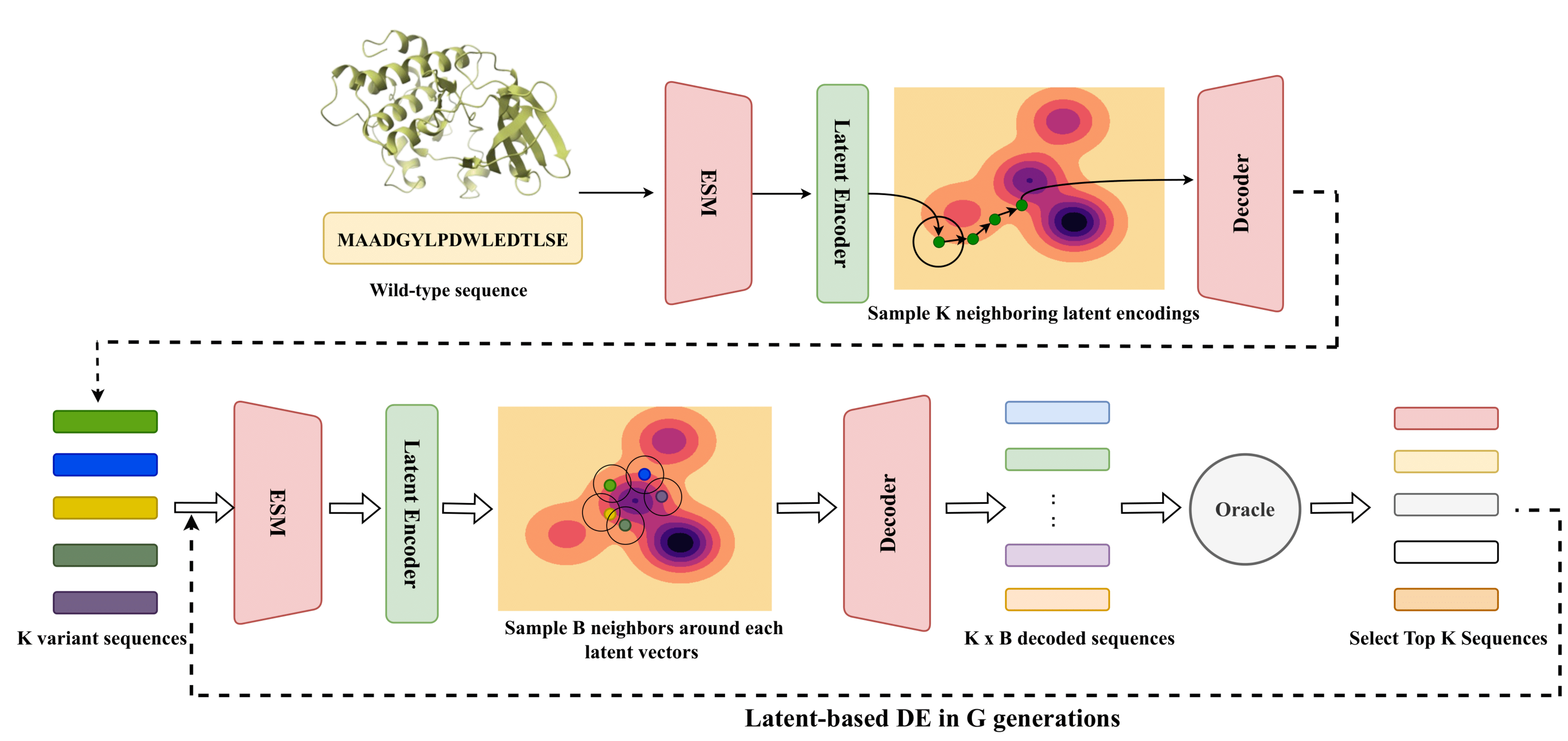
Latent-based Directed Evolution accelerated by Gradient Ascent for Protein Sequence Design

Thanh V. T. Tran[*,1] Nhat Khang Ngo[*,1] Viet Thanh Duy Nguyen[1] Truong Son Hy[2]

[1]FPT Software AI Center, Vietnam
[2]University of Alabama at Birmingham, United States

Motivation





Method

Algorithm 1 Latent-based directed evolution accelerated by gradient ascent

Input: x^{wt}, encoder ϕ, decoder θ, fitness predictor f, # iterations T, # generations G, beam size B, oracle $\mathcal{O}$.

```
P_0 ← ∅
for i = 1 to K do
    Sample z ~ N(μ_φ(x^wt), σ_φ(x^wt)^2)
    z_T ← gradient_ascent(z, f, T) in Equation (4)
    x̂ ← θ(z_T)
    P_0 ← P_0 ∪ {(x̂, O(x̂))}
end for
for g = 1 to G do
    P_g ← ∅
    for k = 1 to K do
        for b = 1 to B do
            Sample z_{k,b} ~ N(μ_φ(x_k), σ_φ(x_k)^2)
            p ~ U[0, 1)
            if p > threshold then
                Inject noise to z_{k,b} based on Equation (5)
            end if
            x̂_{k,b} ← θ(z_{k,b})
            P_g ← P_g ∪ {(x̂_{k,b}, O(x̂_{k,b}))}
        end for
        P_g ← P_g ∪ P_{g−1}
        P_g ← topK(P_g)
    end for
end for
```

Results

Maximum fitness score and Impact of GA and DE

	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I
x^{wt}	1.408	-6.778	-0.015	0.774	-2.789	-1.260	0.014	-0.262
AdaLead	3.323	-1.545	0.248	-0.373	-1.483	-0.047	0.382	2.765
DyNA PPO	5.331	-2.817	0.570	-0.575	-2.790	-0.060	0.183	2.630
CbAS	5.187	-2.800	0.481	-0.658	-1.784	-0.056	0.276	2.693
CMA-ES	5.125	-3.267	0.590	-0.658	-2.790	-0.086	0.254	2.527
COMs	3.544	-3.533	0.472	-0.860	-20.182	-0.087	0.156	2.086
PEX	3.796	2.378	0.252	4.317	-0.364	0.009	1.326	3.578
GFN-AL	5.028	-4.444	0.654	-0.831	-37.360	-5.738	1.399	3.850
GGS	3.368	2.442	1.121	-1.147	-3.364	-0.972	0.059	4.101
LDE (ours)	**8.058**	**2.636**	**1.745**	**5.120**	-0.103	**0.018**	1.548	**4.297**
– w/o GA	6.407	2.148	1.220	4.597	**-0.099**	-0.531	**1.592**	3.254
– w/o DE	3.677	0.919	-0.024	3.052	-0.701	-1.597	0.285	0.766

Integration of active learning

Algorithm 2 Active Learning with Latent-Based DE

Input: a VAE $M = (\phi, \theta, f)$, training dataset $\mathcal{D}_t$, number of rounds N, optimization oracle $\mathcal{O}$, number of epochs n.

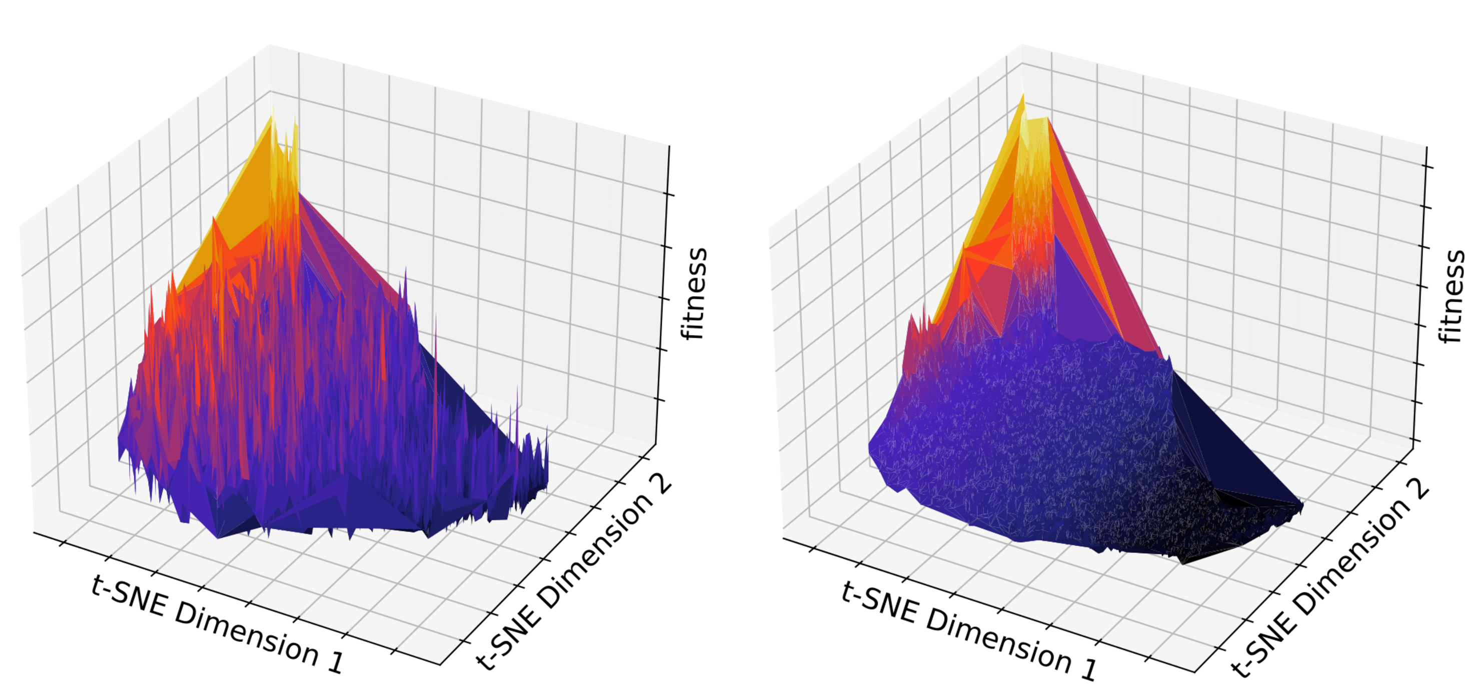
```
Train M on D_t
D_0 ← ∅
for i = 1 to N do
    Run Algorithm 1 with oracle O and M to
    find population P = {(x, O(y)) | x ∈ V^L, y ∈ R}.
    D_i ← D_{i−1} ∪ RemoveDuplicate(P)
    Update M on the data D_i in n epochs
end for
Return P_G
```

Model	TEM	AMIE	LGK	UBE2I
LDE	1.095	-0.558	-0.005	2.976
– w/ active	**2.167**	**-0.015**	**0.022**	**3.698**

Active learning helps with small datasets!!

Latent space visualization

VAE regularization helps organizing latent representations by fitness scores!!



(a) Ground-truth (b) Regularized

Scores with non-autoregressive and autoregressive decoder

	avGFP	AAV	TEM	E4B	AMIE	LGK	Pab1	UBE2I	Average
Non-autoregressive LDE	3.733	1.368	1.095	3.123	-0.558	-0.005	0.089	2.976	1.430
(Autoregressive) LDE	**8.058**	**2.636**	**1.745**	**5.120**	**-0.103**	**0.018**	**1.548**	**4.297**	**3.204**

Correspondence to thy@uab.edu

NEURAL INFORMATION PROCESSING SYSTEMS